

UNIVERZITA HRADEC KRÁLOVÉ
FAKULTA INFORMATIKY A MANAGEMENTU
KATEDRA INFORMATIKY A KVANTITATIVNÍCH METOD

Orchestrace a management virtuálních síťových
funkcí

DIPLOMOVÁ PRÁCE

Autor: Bc. Ondřej Smola

Studijní obor: Aplikovaná informatika

Vedoucí práce: Ing. Vladimír Soběslav, Ph.D.

Hradec Králové

duben, 2016

Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a uvedl jsem všechny použité prameny a literaturu.

V Hradci Králové dne 11. dubna 2016

Ondřej Smola

Poděkování

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean placerat. Duis pulvinar. Maecenas lorem. Mauris tincidunt sem sed arcu. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt.

Anotace

Tato diplomová práce pojednává o aktuálním tématu, kterým je Virtualizace síťových funkcí (Network function virtualization).

Annotation

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean placerat. Duis pulvinar. Maecenas lorem. Mauris tincidunt sem sed arcu. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Phasellus rhoncus. Praesent vitae arcu tempor neque lacinia pretium. Mauris suscipit, ligula sit amet pharetra semper, nibh ante cursus purus, vel sagittis velit mauris vel metus. Etiam posuere lacus quis dolor. Curabitur bibendum justo non orci. Praesent in mauris eu tortor porttitor accumsan. Nullam lectus justo, vulputate eget mollis sed, tempor sed magna. Donec quis nibh at felis congue commodo. Integer tempor. Maecenas libero.

Obsah

1	Úvod	1
2	Základní problematika virtualizace síťových funkcí	3
2.1	NFV a Cloud Computing	5
2.1.1	Distribuční modely	6
2.1.2	Modely nasazení	7
2.2	NFV a SDN	8
2.3	Architektura NFV a VNF	9
2.3.1	Infrastruktura NFV	9
2.3.2	Virtuální síťová funkce	11
2.3.2.1	Service Chaining	12
2.3.3	Management a orchestrace NFV	13
2.3.3.1	Orchestrator	14
2.3.3.2	VNF manager	14
2.3.3.3	Virtualised Infrastructure Manager	14
2.4	Možné technologie pro řešení	14
2.4.1	Hypervisory	14
2.4.2	VNF	14
2.4.3	Cloud platforma	14
2.4.4	SDN	15
3	Popis navrženého řešení a použitých technologií	16
3.1	Architektura navrženého řešení	16
3.2	OpenStack	16
3.3	OpenContrail	16
3.4	Heat Templates	17
3.4.1	FwaaS template	17
3.4.2	LbaaS template	17
4	Testování navrženého řešení	19
4.1	Testovací topologie	19
4.2	Testované síťové funkce	19
4.3	Testování LbaaS	20
4.4	Testování FwaaS	21
4.4.1	Fortigate VM	22
4.4.2	PFSense	22

5	Shrnutí poznatků	26
6	Závěr	27
	Literatura	28
	Přílohy	I

1 Úvod

V dnešní době dochází v datových centrech k nasazování nových moderních technologií. Jednou z nich je například virtualizace a to především v oblasti výpočetního výkonu a úložišť. Je již běžnou praxí, že v datových centrech vše běží na jedné fyzické infrastruktuře, která je abstrahovaná na jeden souvislý blok výpočetního výkonu a jeden souvislý blok úložiště. Dalším takovýmto funkcionálním blokem v datových centrech jsou počítačové sítě. Avšak v počítačových sítích byl, oproti dvěma zmíněným oblastem, pomalejší vývoj inovací a není zde tolik využívána virtualizace. Pro zvýšení efektivity je proto nutné, aby se počítačové sítě staly programovatelnými a mohli být spravovány z jednoho centrálního místa.

Dnes je však zatím síťové funkčnosti soustředěno ve fyzických proprietárních zařízeních jako jsou routery, firewally či load balancery. To znamená, že provozovatelé počítačových sítí se při spouštění nových síťových služeb musí na tyto zařízení spoléhat. Což může vést k zdlouhavému nasazování, zvýšené spotřebě energií a investici do školení pracovníků pro dané proprietární zařízení. Zároveň zde není možnost, aby síť mohla být dynamicky ovládána dle aktuálních požadavků uživatelů sítě. Například vývojář nemůže hned nasadit aplikaci do produkce. Musí nejprve čekat na síťový tým než patřičně nakonfiguruje síťové prvky pro správné a bezpečné fungování celé infrastruktury.

Virtualizace síťových funkcí se zaměřuje na transformaci způsobu, jakým síťový architekti přistupují k oblasti počítačových sítí a to pomocí stávajících a neustále se vyvíjejících virtualizačních technologií. Snaha je tedy přesunout mnoho typů síťového příslušenství z fyzických síťových prvků do standardních průmyslově používaných serverů a úložišť, které mohou být umístěny v datových centrech či přímo u koncových zákazníků. Tímto lze dosáhnout virtuálních síťových funkcí, které mají naprosto stejnou funkcionalitu jako síťové funkce umístěné v síťových prvcích.

Cílem této diplomové práce je analyzovat aktuální stav v oblasti virtualizace síťových funkcí. Dále je cílem navrhnout jednoduchý framework pro virtualizaci síťových funkcí spolu s ukázkou několik příkladů síťových funkcí. Celý framework by měl sloužit k možnosti rychlého a jednoduchého nasazení vybraných síťových funkcí. Současně by k jeho vytvoření měli být použity aktuálně dostupné technologie. Toto řešení musí být univerzální, nezávislé na vendorech a flexibilní.

Celá struktura této práce je rozdělena na několik částí. V druhé kapitole jsou vysvětleny hlavní pojmy a problematika oblasti virtualizace síťových funkcí. Třetí je popisu návrhu řešení frameworku pro virtualizaci síťových funkcí a popisu použitých použitých technologií pro tento návrh. Ve čtvrté kapitole je věnována testování a ukázce, jak navržený framework funguje. Na konci této práce dojde k závěrečnému shrnutí.

Závěrečná práce byla zpracována ve spolupráci s firmou tcp cloud a.s., která poskytuje implementace jednoho z nejlepších cloudových řešení na světě. Firma umožnila využít jejich stávající infrastrukturu v nejmodernějším datovém centru v České republice, které je v budově Technologického centra Písek s.r.o.

2 Základní problematika virtualizace síťových funkcí

Jak již vyplývá z názvu, tak tato kapitola se zabývá základní analýzou a popisem problematiky spojené s oblastí virtuální síťových funkcí.

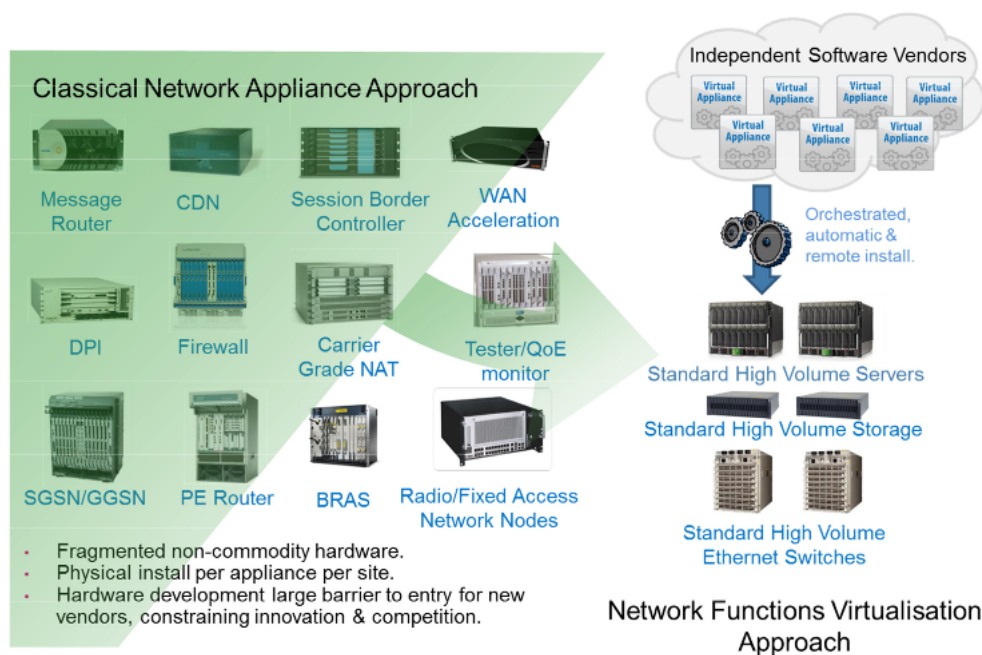
Produkční vývoj v telekomunikačním průmyslu se tradičně řídil přísnými standardy kvůli stabilitě a kvalitě komunikace. Přestože tento model v minulosti fungoval, tak vedl nevyhnutelně k dlouhým produkčním cyklům, pomalému tempu vývoje a spoléhání se na proprietární či specializovaný hardware. S příchodem výrazné konkurence v komunikačních službách, od rychle postupujících organizací operujících ve velkém měřítku na veřejném internetu, podnítil poskytovatele služeb pro hledat nových způsobů, jak změnit dosavadní způsob produkčního vývoje.

Pro vyřešení toho problému bylo navrženo v publikacích [1] a [2] skupinou několika telekomunikačních provozovatelů řešení ve formě virtualizace síťových funkcí (network functions virtualization). Toto řešení má za cíl zlepšit následující aspekty provozu telekomunikačních sítí:

- Smíření investičních nákladů – snížení potřeby nákupu jednoúčelových hardwarových zařízení, možnost platby pouze za využití kapacity a snížení rizik přílišného předimenzování kapacit
- Snížení provozních nákladů – snížení prostoru, napájení a požadavky na chlazení, zjednodušení správy a řízení síťových služeb
- Urychlení Time-to-market – zkrácení doby pro nasazení nových síťových služeb, chopení se nových příležitosti na trhu, vyhovění potřebám zákazníka
- Doručit agilitu a flexibilitu – možnost rychle škálovat (rozšiřovat nebo zmenšovat služby) dle měnících se požadavků od zákazníka. Podpora služeb, které mají být dodány pomocí softwaru na libovolném standardním serverovém hardwaru

Jak je uvedeno v [3] a [4], tak celá myšlenka je založena na tom, že dojde k separování softwarové funkcionality v síťových prvcích od proprietárního hardwaru, na kterém běží. To umožní se síťovými funkcemi zacházet jako s klasickými softwarovými aplikacemi, které mohou běžet na standardním komerčně dostupných serverech

jenž organizace v současnosti používají. Tím bude zároveň umožněno flexibilní nasazování těchto síťových funkcí a jejich dynamický provisioning. Díky tomu, že jsou síťové funkce odděleny od hardwaru, tak je také možné jejich vhodnější umístění v topologii. To znamená dle požadavků na umístění mohou být nasazeny v datových centrech, síťových uzlech či přímo v uživatelských koncových bodech. Hlavní koncept virtualizace síťových funkcí znázorňuje obrázek č. 2.1.



Obrázek 2.1: Koncept virtualizace síťových funkcí (NFV)

Z zmínění stojí poznámka v [3], kde je řečeno, že obecný koncept oddělení síťové funkce od hardwaru ještě nutně neznamená potřebu využití virtualizace. Protože budou síťové funkce dostupné jako software, tak mohou být nainstalovány a provozovány přímo na fyzickém stroji. Ovšem rozdíl je, že tento stroj již nebude speciální hardware, ale klasický server. Tento scénář může být do jisté míry použit při nasazování síťových funkcí v malém měřítku např. v uživatelských koncových bodech. Avšak pro plné využití všech výše zmíněných výhod, které jsou třeba ve velkých datových centrech, je třeba s použitím virtualizace počítat. To vše umocňuje fakt, že většina datových center v současnosti již využívá cloud computing.

Pro lepší pochopení a přehlednost celé této práce zde budou rozlišeny následující pojmy, se kterými se lze také setkat v odborné literatuře a které budou dále v této práci používány.

- Síťová funkce (Network function - NF) - Toto je komponenta síťové infrastruktury, která má dobře definované funkční chování, jako například směrování, NAT,

Load balancing, Intrusion detection, atd.

- Virtuální síťové funkce (Virtual network function - VNF) - Je stejná jako NF, ale zde je funkčnost implementována pomocí softwaru a je nezávislá na hardwaru, na kterém běží.
- Virtualizace síťových funkcí (Network Functions Virtualization - NFV) - Zde se jedná o označení celého konceptu či frameworku.

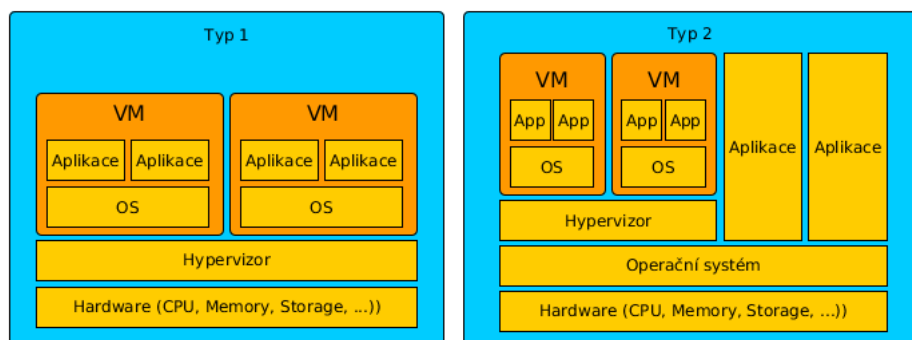
2.1 NFV a Cloud Computing

Cloud Computing a virtualizace obecně jsou hlavní technologie, které umožnily virtualizaci síťových funkcí. Z toho důvodu zde budou stručně představeny.

Virtualizací obecně označujeme techniky, které umožňují k dostupným hardwarovým zdrojům přistupovat jiným způsobem, než jakým fyzicky existují. Je tomu díky softwaru, který tento hardware abstrahuje a vytvoří tím virtuální prostředí. Virtualizované prostředí se dá snadněji přizpůsobit potřebám uživatelů, případně skrýt pro uživatele nepodstatné detaily (jako např. rozmístění hardwarových prostředků). Tento software se nazývá hypervisor.[5]

Jak zmiňuje [6], tak existují tyto dva základní typy hypervisorů:

- Typ 1 (Nativní) - Tento hypervisor běží přímo na fyzickém hardwaru. Tím umožňuje provozovat více operačních systémů na jednom fyzickém stroji. Příkladem takového hypervisoru je VMware ESXi a XEN.
- Typ 2 (Hostovaný) - Na rozdíl od předchozího případu tento typ hypervisoru běží v prostředí operačního systému. Příkladem je například KVM či Microsoft Hyper-V



Obrázek 2.2: Schéma hypervisorů

Obrázek 2.2 zobrazuje schématický popis obou hypervisoru a jejich rozdíl. Problematika virtualizace je velice rozsáhlá a více informací o ní poskytují zdroje [5] a [6].

Cloud Computing je nejpokročilejší formou virtualizace, kterou v poslední době začala provozovat většina větších organizací, jak ukazuje [7]. Cloud Computing má mnoho definic. V [8] je uvedeno, že se jedná o model, ve kterém lze uchovávat a poskytovat výpočetní zdroje, které jsou dostupné pomocí počítačové sítě s tím, že uživatelé k nim mohou přistupovat vzdáleně. Virtualizace síťových služeb umístěná v datových centrech je přímo svázaná s touto technologií. Z tohoto důvodu zde bude podrobněji popsán jejich vztah.

2.1.1 Distribuční modely

Dle [9] lze cloudové služby rozdělit do 3 základních kategorií. V [10] jsou k těmto kategoriím přiřazeny příklady užití z oblasti virtualizace síťových funkcí.

- **Infrastructure as a Service (IaaS)** - Nejzákladnější model poskytování cloudových služeb. IaaS cloudové platformy nabízejí například výpočetní výkon, virtuální disky, blokové a souborové úložiště či virtuální síť. Poskytovatelé IaaS cloudových platform poskytnou tyto zdroje na vyžádání ze svých datových center. Toto je možné díky skupině hypervisorů v rámci cloudu, které mohou provozovat velké množství virtuálních strojů a mají schopnost škálovat poskytované služby v závislosti na měnících se požadavcích přicházejících od zákazníků. Tento model může tedy sloužit i pro poskytnutí všech potřebných zdrojů celé infrastruktury pro virtualizaci síťových prvků, neboli Network Function Virtualization Infrastructure as a Service. Zde má uživatel pod nejvíce možností, jak navrhnout a spravovat virtuální síťové funkce, protože zásadě dokáže nasazovat i vlastně navržené síťové funkce a nejen ty, které mu poskytuje provozovatel cloudu.
- **Platform as a Service (PaaS)** - V modelu Platforma jako služba (PaaS) hostují poskytovatelé cloudových služeb určitou počítačovou platformu, kterou následně poskytují koncovým uživatelům přes Internet. Tato platforma většinou bývá prostředím nějakého operačního systému, prostředí pro běh určitého programovacího jazyka, databáze a webový server. Vývojáři aplikací tím pádem mohou provozovat a případně vyvíjet svá softwarová řešení bez výrazných nákladů a složitého nákupu a konfiguraci potřebného hardwaru a softwaru. Některé PaaS platformy nastavuje výpočetní a úložné prostředky aplikace automaticky tak, aby odpovídala aktuálním požadavkům aplikace bez nutnosti zásahu zákazníka. NVF v tomto modelu může nabízet síťové služby, které se mohou skládat z více virtuálních síťových funkcí, neboli Virtual Network Platform as a Service. Zde je poskytnuta uživateli velká kontrola nad konfigurací a ovládáním celé platformy.

- Software as a Service (SaaS) - V modelu SaaS provozují poskytovatelé cloudových služeb aplikační software v cloudu a uživatelé k tomuto softwaru přistupují pomocí klientského software (např. webové prohlížeče). Uživatelé cloudu tedy nespravují infrastrukturu ani platformu, kde aplikace běží. Není proto třeba zde nic instalovat a spouštět aplikace na vlastních počítačích uživatele, což velmi zjednodušuje údržbu. Cloudové aplikace se liší od ostatních aplikací v možnostech škálování, kterého může být dosaženo díky distribuci úkolů na více virtuálních strojů, a tím reagovat na měnící se poptávku. Tento proces je pro uživatele služby transparentní, uživatel vidí pouze jeden přístupový bod pro danou aplikaci. Do této kategorie služeb může patřit poskytování virtuálních síťových funkcí, která je pouze ve formě softwarové aplikace, neboli VNaaS. Takovéto aplikace poskytují síťovou funkci pro síťové správce a uživatele nejčastěji v privátním cloudu.

2.1.2 Modely nasazení

Existuje několik základních modelů nasazení cloud computingu resp. cloudových platforem, které uvádí [9]. V [10] lze k nim opět najít určité příklady z oblasti virtualizace síťových funkcí.

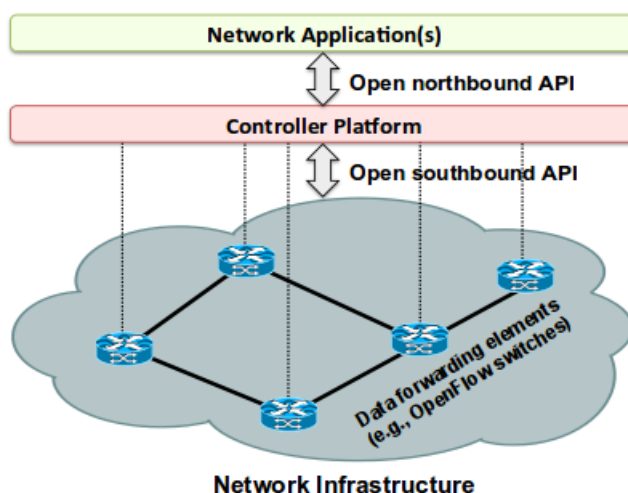
- Privátní cloud - Privátní cloud je infrastruktura provozována výhradně v rámci jedné organizace. Může být spravován interně nebo prostřednictvím třetí strany a hostování může být opět interní nebo externí. Aby mohl podnik využít privátní cloud, musí nejprve navrhnout a uzpůsobit k tomuto účelu svoji stávající infrastrukturu, která musí být virtualizována. Vlastní přechod vyvolává řadu bezpečnostních otázek, které je třeba řešit, aby se zabránilo vážným zranitelnostem celého řešení. Privátní cloud je přesně ten typ modelu, kde lze najít využití pro virtualizaci síťových funkcí.
- Veřejný cloud - Veřejné cloudové jsou cloudové služby, jako jsou aplikace, výpočetní výkon, úložiště a další, které jsou k dispozici široké veřejnosti. Služby jsou poskytovány zdarma nebo podle modelu platby za množství použitých služeb. Je zvykem, že veřejní poskytovatelé cloudových služeb, jako je Amazon AWS, Microsoft nebo Google, vlastní a provozují hardwarovou infrastrukturu a nabízejí k ní přístup pouze přes Internet. V tomto modelu není očekáváno využívání NFV.
- Hybridní cloud - Hybridní cloud je spojení dvou nebo více cloudů (soukromých, komunitních nebo veřejných), které zůstávají samostatné, ale jsou těsně propojeny. Toto složení rozšiřuje možnosti nasazení cloudových služeb a tím umožňuje

IT organizacím využít veřejné cloudové prostředky k uspokojení dočasných potřeb. Tato schopnost umožňuje hybridním cloudům škálovat přes více nezávislých cloudů. V tomto modelu může být využito NFV především na straně soukromého cloudu.

- Komunitní cloud - V rámci komunitního cloudu sdílí infrastrukturu cloudu několik organizací, které mají společné zájmy (bezpečnost, dodržování předpisů, působnost, atd.). Komunitní cloud může být spravován interně nebo prostřednictvím třetí strany. Náklady jsou rozloženy mezi méně uživatelů než na veřejném cloudu. V tomto modelu může být využito NFV, pokud se provozovatelé takového cloudu domluví.

2.2 NFV a SDN

Softwarově definované sítě (SDN) je další z nových technologií, která se snaží vylepšit a automatizovat správu stávajících počítačových sítí. Dle [11] jde o koncept, ve které je oddělena řídicí logika (control plane) z jednotlivých routerů a switchů, které přeposílají traffic (data plane). Tím, že dojde k oddělení datové a řídicí vrstvy, se routery a switche stanou pouze přeposílající data a veškerá řídicí logika může být implementována v jednom logicky centrálním místě (SDN Controller). Z tohoto centrálního místa lze do jednotlivých routerů a switchů předávat instrukce pomocí aplikačních programovacích rozhraní (API). Samotný SDN Controller také obsahuje API, které mohou využívat aplikace a tím řídit, resp. programovat celou počítačovou síť. Obrázek č. 2.3 ukazuje jednoduché schéma softwarově definovaných sítí.



Obrázek 2.3: Schéma SDN, převzato z [11]

Přestože Softwarově definované sítě a virtualizace síťových funkcí jsou dvě různé technologie a koncepty, tak se navzájem se doplňují. Fakt, že SDN umožňuje programaticky ovládat počítačovou síť, lze využít pro poskytnutí programovatelné konektivity mezi jednotlivými virtuálními síťovými funkcemi. Naopak SDN může využít NFV tím, že implementuje potřebné síťové funkce jako software. Může tak virtualizovat SDN Controller, který tak může běžet na co nejvýhodnějším místě v datovém centru. Je vidět, že tyto dvě technologie se dobře doplňují, proto jsou často součástí jednoho řešení. [12]

2.3 Architektura NFV a VNF

V předchozí sekci byla popsána myšlenka a motivace související s virtualizací síťových funkcí. Protože cílem této práce je navržení jednoduchého NFV frameworku, tak je nejprve nutné se seznámit s jeho obecnou architekturou. V této práci se bude vycházet z referenční architektury NVF [13], která byla navržena organizací ETSI. Jedná se pouze o funkční návrh bez náznaků konkrétní implementace. Od této skupiny existují i podrobnější návrhy jednotlivých částí celého NFV frameworku, které v této práci budou také popsány v příslušných kapitolách.

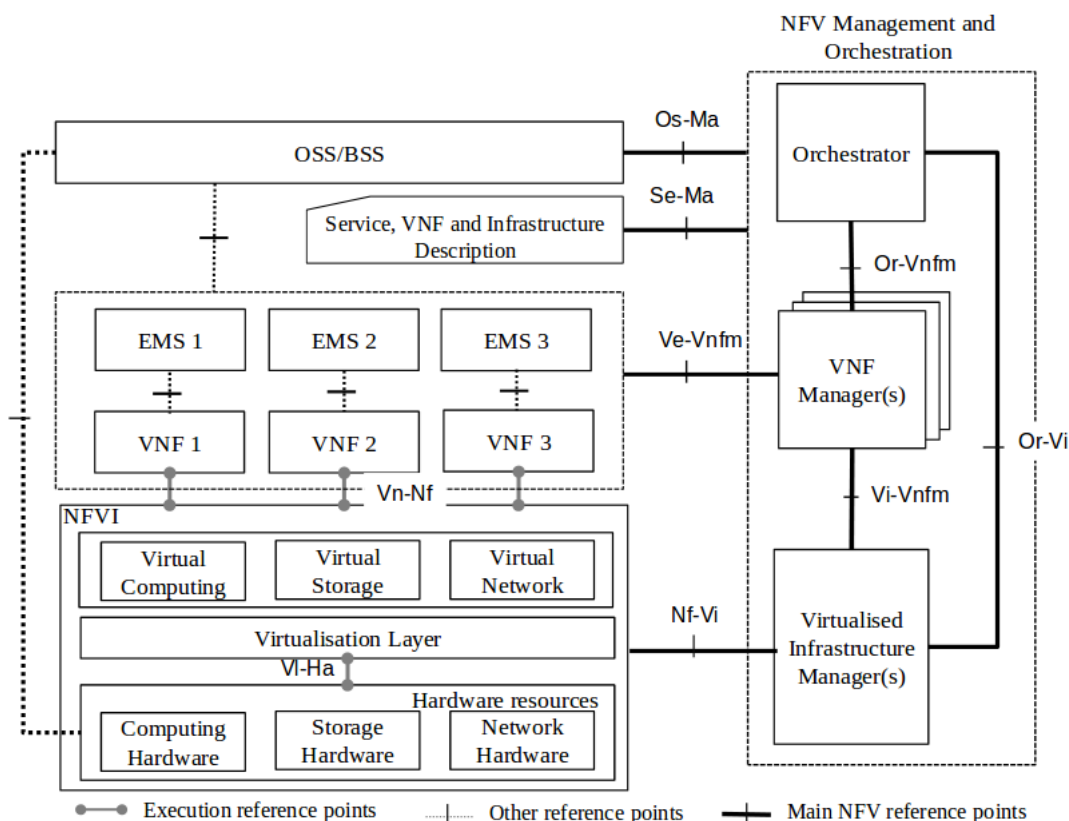
Jak je vidět na obrázku č. 2.4, tak celá architektura se dá rozdělit na tyto 3 hlavní části:

- **Infrastruktura virtualizace síťových funkcí (NFVI)** - Jsou všechny softwarové a hardwarové zdroje potřebné k vytvoření prostředí, ve které mohou být jednotlivé VNFs být nasazeny. Tato infrastruktura může být velice rozsáhlá, proto je její součástí i síť poskytující konektivitu mezi vzdálenými lokacemi infrastruktury.[14]
- **Virtualizované síťové funkce (VNFs)** - Jsou softwarové implementace síťových funkcí, jako je např. NAT a routing, které mohou být nasazeny na NFV infrastruktuře.
- **Management a orchestrace NFV (NFV-MANO)** - zde se jedná o řízení softwarových a hardwarových zdrojů v celé infrastruktuře NFV a životního cyklu jednotlivých virtuálních síťových funkcí. Tato část se tedy zaměřuje na řízení a správu všech úloh související v virtualizaci v NFV frameworku. [14]

Tyto funkční bloky se ještě dále dělí, proto dále v této práci budou tyto jednotlivé části popsány podrobněji a současně k nim budou uvedeny různé možnosti řešení.

2.3.1 Infrastruktura NFV

Ve zdroji [15], který detailně popisuje infrastrukturu pro virtualizaci síťových funkcí (NFVI), je uvedeno, že je v ní sdružení všech základních zdrojů potřebných pro běh



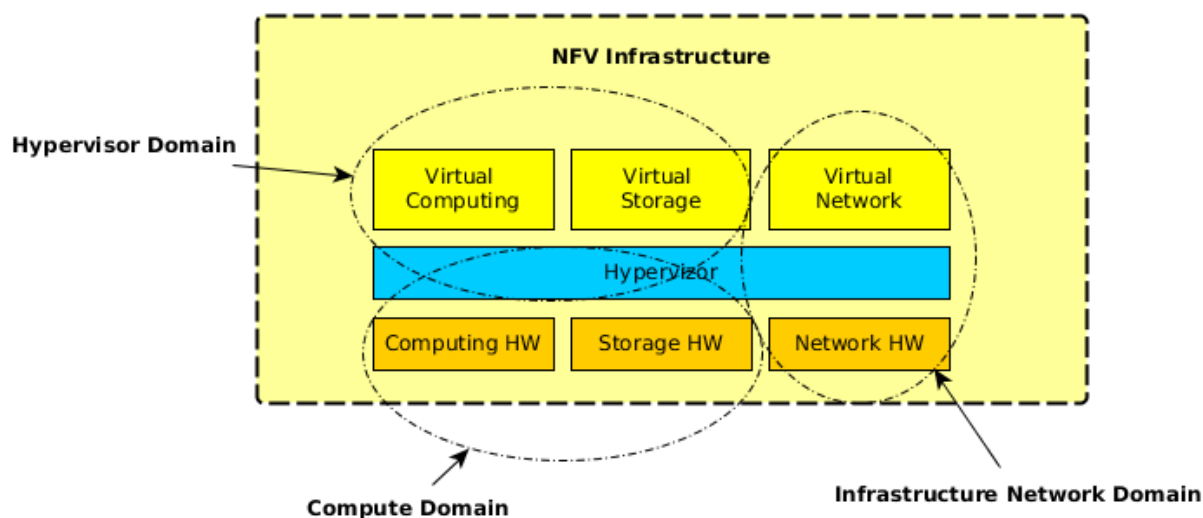
Obrázek 2.4: NFV architektura, převzato z [13]

virtuálních síťových funkcí (VNF). Z tohoto důvodu sem patří veškerý hardware. Do NFVI také patří některé softwarové komponenty, které jsou společné mnoho VNF a poskytují funkcionalitu potřebnou pro podporu nasazení, propojení či managementu VNF. Celou infrastrukturu může tvořit jeden či více strojů, které mají tyto potřebné funkce. Tyto stroje také mohou být umístěny v různých spolu spojených lokacích.

Pro zjednodušení lze celou NFV infrastrukturu rozdělit do 3 následujících domén:

- **Compute Domain** - Do této domény patří veškeré hardwarové zdroje jako jsou servery, úložiště a komponenty, které tyto zdroje obsahují, např. procesory, pevné disky, síťové karty, atd. Zároveň je zde řešen návrh fyzické topologie. [16]
- **Hypervisor Domain** - Toto je doména, které představuje softwarové prostředí abstrahující hardware v compute doméně a poskytuje je jako virtuální zdroje. Tyto zdroje následně mohou využívat virtuální síťové funkce. [17]
- **Infrastructure Network Domain** - V této doméně je řešeno veškeré propojení výše zmíněných domén. Tedy fyzické i virtuální infrastruktury. [18]

Funkci obsaženou v jednotlivých doménách znázorňuje obrázek č. 2.5. Více informací na tuto problematiku lze nalézt v [15] a ve zdrojích uvedených u každé domény.



Obrázek 2.5: Schéma NFV infrastruktury

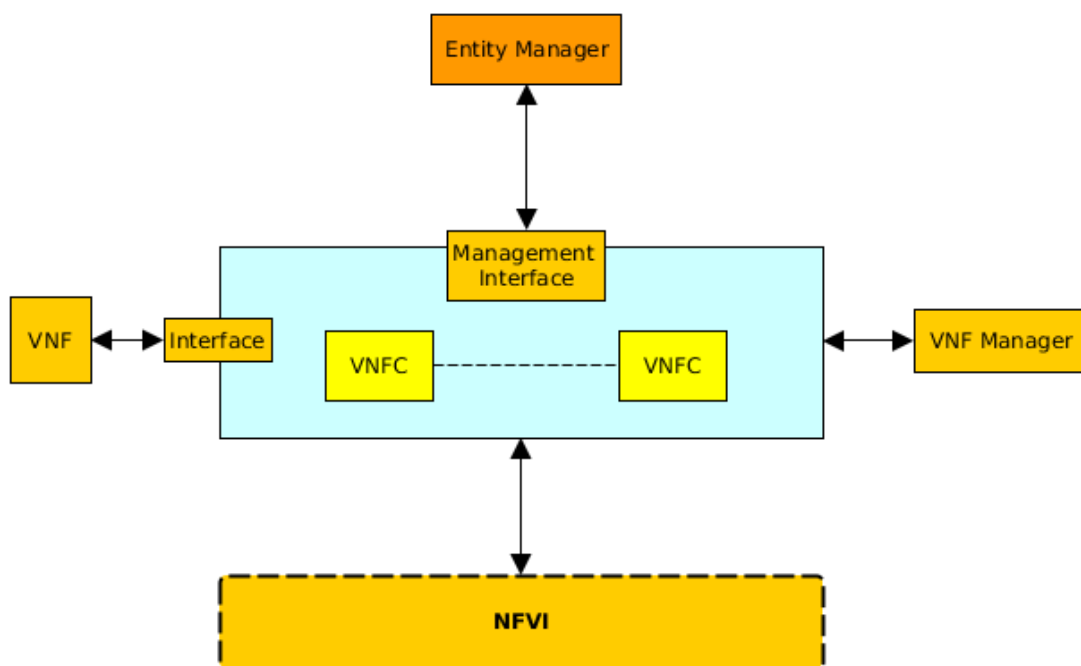
Dá se říci, že návrh infrastruktury pro NVF je podobný jako pro návrh infrastruktury pro cloud computing platformu. Měl by se tedy skládat z generických a komerčně vysoce dostupných serverů, které by měli být zapojeny do switche a tím by měla být zajištěna konektivita. Na tyto servery je následně nasazen jeden z dostupných hypervisorů. Výběr správného hypervisoru, které jsou v současné době dostupné na trhu, je hlavní podmínka správného a funkčního návrhu této části NFV frameworku. Přehled hypervisorů je uveden v kapitole 2.4.1. V produkčním prostředí by součástí řešení bylo samozřejmě řešení síťového návrhu. Tato práce však má sloužit pouze jako ukázka a z tohoto důvodu zde nebude síťový návrh zmíněn.

2.3.2 Virtuální síťová funkce

Virtuální síťová funkce (VNF) je dle [19] určitá síťová funkce, která běží na NVF infrastruktuře a je zároveň NVF frameworkem řízena a spravována. Zároveň musí mít dobře definované rozhraní k ostatním síťovým funkcím, k VNF Managerovi a měla by obsahovat management rozhraní či port.

Síťové funkce jsou tedy obvykle realizovány pomocí virtuálních strojů či dnes již často používaných kontenerů. Závisí na použité virtualizační platformě. Jedna VNF může být obsažena v jednom virtuálním stroji nebo může být roztažena přes více virtuálních strojů.

Na obrázku č. 2.9 je vidět jednoduché schéma virtuální síťové funkce. Celý životní cyklus VNF, což je vytvoření, spuštění, zastavení, smazání a škálování, řídí VNF Manager, který je součástí NVF managementu a orchestrace. Současně je možné dynamicky změnit aktuální konfiguraci pomocí Entity manageru (EM) přes management interface. EM může spravovat více VNF nebo právě jednu. Vnitřní struktura celé instance může být tvořena více komponentami (VNFC), které spolu mohou být navzájem provázány. Toto provázání však nemusí být viditelné zvenčí.



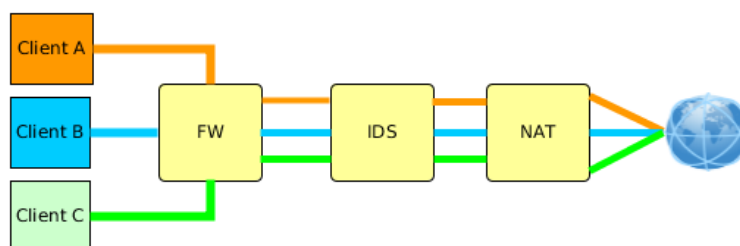
Obrázek 2.6: Schéma virtuální síťové funkce

2.3.2.1 Service Chaining

Jednou z výhod NFV je možnost využít Service Chaining. Service chaining je ve skutečnosti součástí SDN, které bylo zmíněno v kapitole 2.2. Jde o princip jakým lze dynamicky pospojovat jednotlivé VNF. [12]

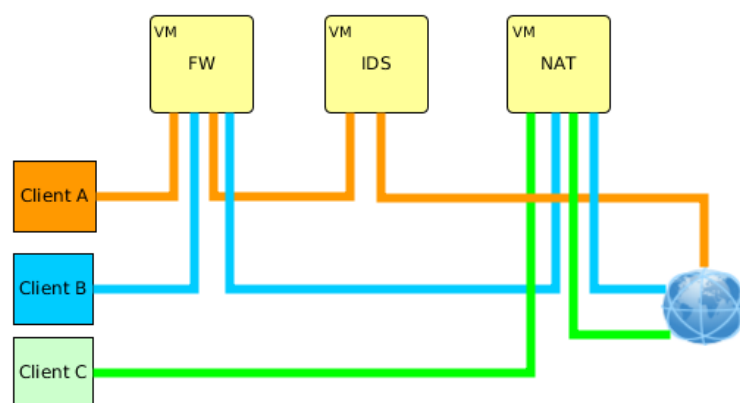
Service chaining není ve skutečnosti nic nového. V klasických počítačových sítích je používán také, ale pomocí fyzických síťových prvků. Jedná se zjednodušeně o způsob zapojení mezi jednotlivými síťovými prvky (či VNF) a způsob, jakým na sebe navazují. Příklad takového zapojení je vidět na obrázku č. 2.7. Zde se provozovatel sítě rozhodl, že odchozí data z klientských stanic musí jít přes firewall, IDS a nakonec přes NAT do Internetu. Příchozí data mají logicky obrácené pořadí. Toto zapojení funguje dobře pro síť, kde není třeba rozlišovat cestu jakou proudí data jednotlivých uživatelských stanic.

Ale není to optimální řešení pro síť s více uživateli, kde každý požaduje jinou síťovou funkci. Potřeba jednotlivých síťových služeb se samozřejmě může v čase měnit. Příklad takové sítě lze nalézt ve většině datových center.



Obrázek 2.7: Ukázka klasického service chainigu pomocí fyzických síťových prvků

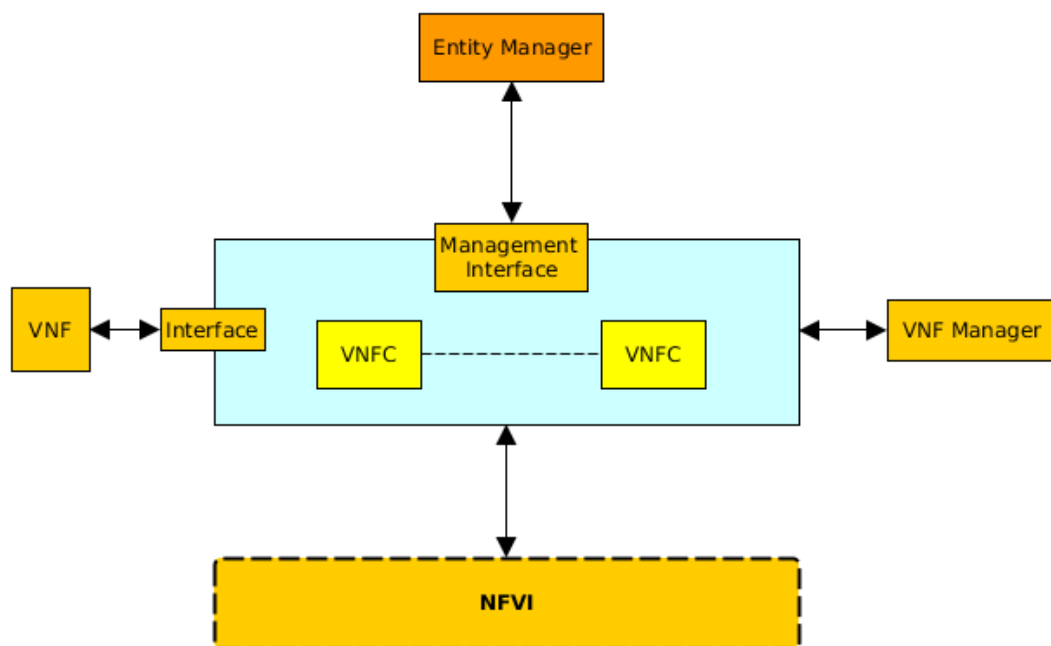
Zde tedy přichází na řadu VNF spolu s SDN. Protože jednotlivé VNF existují jako virtuální stroje, tak mohou být dynamicky nasazovány dle aktuálních požadavků jednotlivých klientů a pomocí SDN mohou být tyto VM dynamicky pospojovány. Obrázek č. 2.8 ukazuje schéma zapojení, kde každý klient může mít jinou požadovanou cestu do internetu. Je možná i varianta, kde každý klient má své vlastní VNF s jinou konfigurací.



Obrázek 2.8: Ukázka VNF service chainigu

2.3.3 Management a orchestrace NFV

Management a orchestrace virtualizace síťových funkcí (NFV MANO) je nejdůležitější celého NFV frameworku.



Obrázek 2.9: Schéma NFV MANO

2.3.3.1 Orchestrator

2.3.3.2 VNF manager

2.3.3.3 Virtualised Infrastructure Manager

2.4 Možné technologie pro řešení

2.4.1 Hypervisory

KVM, Hyper-V, VMware ESXi

2.4.2 VNF

Juniper vSRX, Fortigate VM, PFSense, Cisco ASA, Vyatta, M0n0wall, OpenStack Neutron HAproxy

2.4.3 Cloud platforma

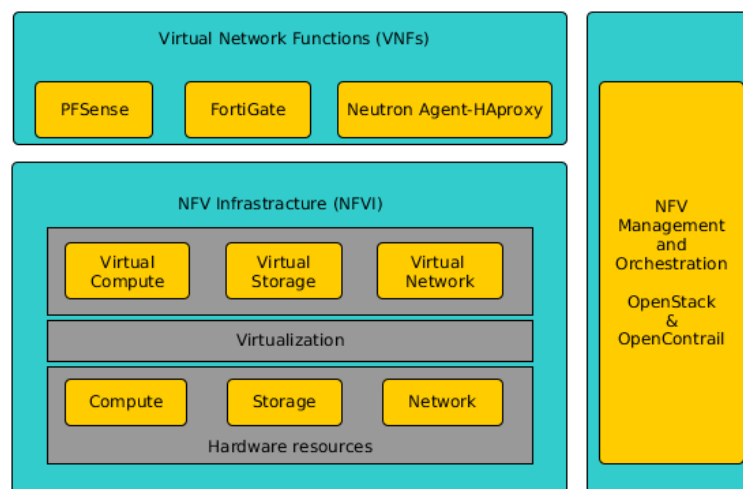
OpenStack, VMware

2.4.4 SDN

OpenContrail, VMware

3 Popis navrženého řešení a použitých technologií

3.1 Architektura navrženého řešení



Obrázek 3.1: Architektura NFV řešení

3.2 OpenStack

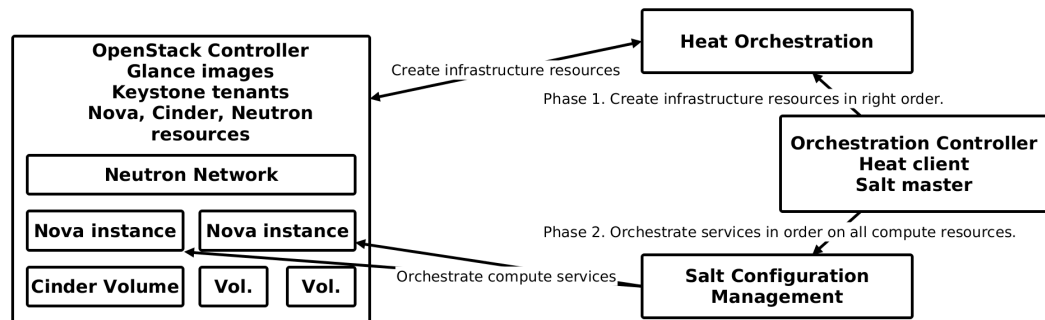
Popis Openstacku

OpenStack Heat Templates are used to demonstrate load balancing and firewalling inside of Openstack.

3.3 OpenContrail

Popis OpenContrailu.

3.4 Heat Templates



Obrázek 3.2: Popis heat orchestrace

Popis co jsou to heat templates.

Heat is the main project of the OpenStack orchestration program. It allows users to describe deployments of complex cloud applications in text files called templates. These templates are then parsed and executed by the Heat engine.

3.4.1 FwaaS template

Pro FwaaS je naruhot heat template, který obsahuje:

- 1 firewall instanci
- 1 testovací instanci
- 1 management instanci
- management síť
- privátní síť
- contrail policy

3.4.2 LbaaS template

Navržený heat template pro LbaaS v sobě obsahuje následující prostředky, které se po spuštění pokusí vytvořit.

- pool
- members
- health monitoring

- 2 web instance
- privatni síť
- public síť

4 Testování navrženého řešení

V předchozí kapitole byly popsány technologie, které byly v této práci použity. V této kapitole bude uvedeno několik příkladů, jak lze jednoduše vytvořit VNF v prostředí OpenStack a OpenContrail pomocí heat templatů. Všechna uvedená řešení byla testována v prostředí OpenStack s OpenContrailem, které bylo pro tyto účely poskytnuto společností tcp cloud a.s.

4.1 Testovací topologie

The NFV topology consist of 5 nodes. The management node is used for public IP access and is accessible via SSH. It is also used as a JUMP host to connect to all other nodes in the blueprint. The controller node is the brains of the operation and is where Openstack and OpenContrail are installed. Finally, we have three compute nodes named Compute 1, Compute 2 and Compute 3 with Nova Compute and the Opencontrail vRouter agent installed. This is where the data plane forwarding will be carried out.

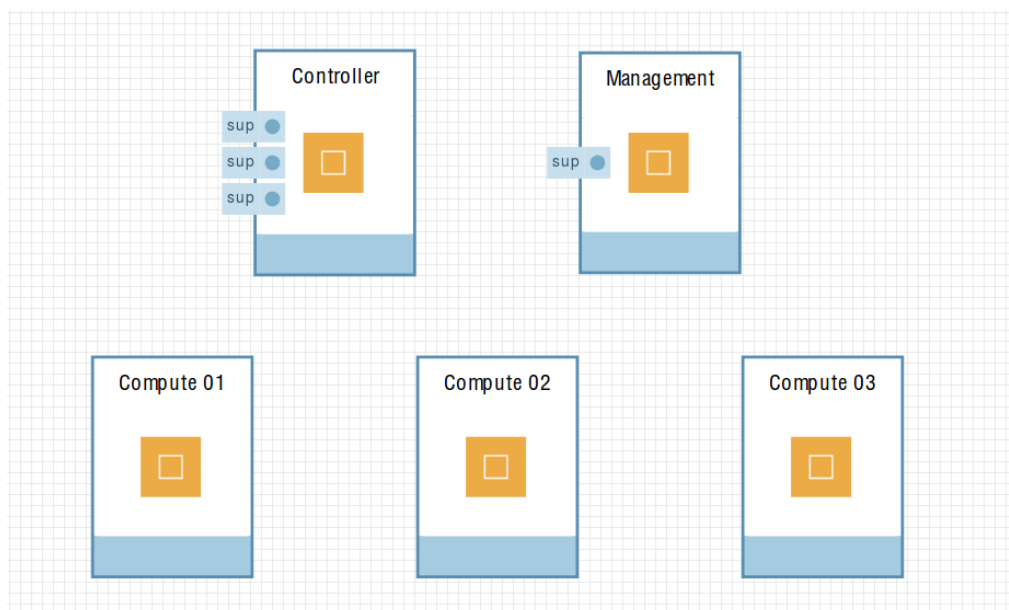
The diagram below display the 5 components used in the topology. All nodes apart from the management node have 8 CPU, 16GB of RAM and 64GB of total storage. The management node has 4 CPU, 4GB of RAM and 32GB of total storage.

4.2 Testované síťové funkce

Navrhnutá řešení v této práci předvádějí virtuální síťové funkce pro firewall a load balancing. Jsou zde ukázány celkem 3 scénáře případu užití. Dva jsou zaměřeny na FwaaS (Firewall as a Service) a jeden na LbaaS (Load balancer as a Service). Všechna řešení jsou vytvořena pomocí Heat templatů, které se spouští v prostředí OpenStack.

Aby mohla být nějaká VNF vůbec vytvořena, tak musel být nejprve zvolen software či operační systém, který má požadovanou funkci implementovanu. Pro tyto účely byly použity následující řešení:

- PFSense – open-souce firewall založený na operačním systému FreeBSD.
- FortiGate-VM – je plnohodnotně vybavený Fortigate firewall zabalený jako virtuální instance.



Obrázek 4.1: Testovací topologie

- Neutron Agent-HAproxy – je velmi rychlé a spolehlivé řešení nabízející vysokou dostupnost, load balancing a proxy pro aplikace založené na TCP a HTTP

Následující diagram znázorňuje logickou architekturu navrženého řešení dle referenční architektury zmíněné v kapitole 2.4. OpenStack spolu s OpenContraiem poskytují NFV infrastrukturu jednotlivé VNF jsou řízeny pomocí Heat.

4.3 Testování LbaaS

Pro vytvoření heat stacku s Load balancerem je nutné daný template vytvořit pomocí příkazu:

```
heat stack-create -f heat/templates/lbaas_template.hot -e heat/env/lbaas_env.env lbaas
```

Tento příkaz vytvoří všechny již uvedené prostředky pro load balancing. Konkrétní load balancer má nakonfigurovanou virtual ip adresu (VIP) a k ní přiřazenou floating adresu, která je přístupná z externích sítí. Zároveň má tento load balancer přiřazený pool, ke kterému je přiřazena privátní síť 10.10.10.0/24. Na obrázku č. X znázorňuje tento pool a obrázek č. X+1 jsou vidět členové (members) toho poolu.

Další zdrojem, který byl vytvořen je health monitor, který lze vidět na obrázku č. X+2. Díky němu má load balancer přehled o aktuálním stavu webových instancí. Pokud by náhodou některá z nich přestala odpovídat, v tomto případě na ping, tak by load balancer na tuto instanci přestal zasílat traffic.

Name	Description	Provider	Subnet	Protocol	Status	VIP
lbass		opencontrail	10.10.10.0/24	HTTP	ACTIVE	lb_pool.vip

Obrázek 4.2: Vytvořený pool

IP Address	Protocol Port	Weight	Pool	Status
10.10.10.100	80	1	lbass	ACTIVE
10.10.10.101	80	1	lbass	ACTIVE
10.10.10.100	80	1	lbass	ACTIVE
10.10.10.101	80	1	lbass	ACTIVE

Obrázek 4.3: Vytvoření members

Finální síťovou topologií znázorňuje obrázek č. X+3.

Otestování webových serverů lze provést příkazem curl, kterému dáme jako parametrem IP VIP nebo floating IP load balanceru. Po několika takovýchto zadání tohoto příkazu je vidět, že oba web servery odpovídají a je probíhá mezi nimi load balancing metodou round robin. Celý tento test je vidět na obr. č. X+4

4.4 Testování Fwaas

Pro vytvoření heat stacku s PFSense z šablony lze použít příkaz:

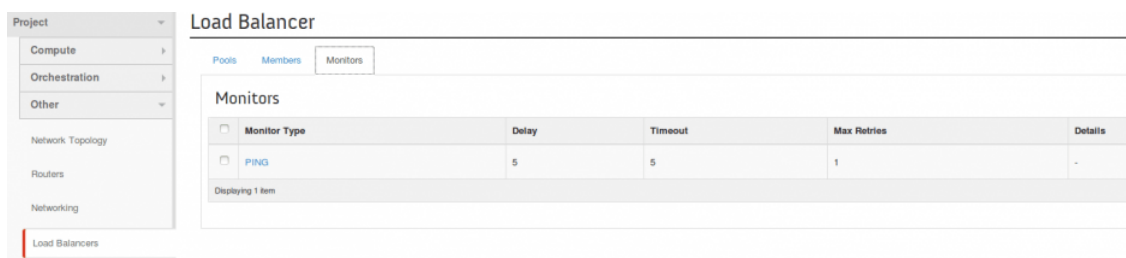
```
heat stack-create -f heat/templates/fwaas_mnmg_template.hot -e heat/env/fwaas_pfsense_env.env pfsense
```

a pro vytvoření heat stacku s Fortigate VM jde vytvořit pomocí příkazu:

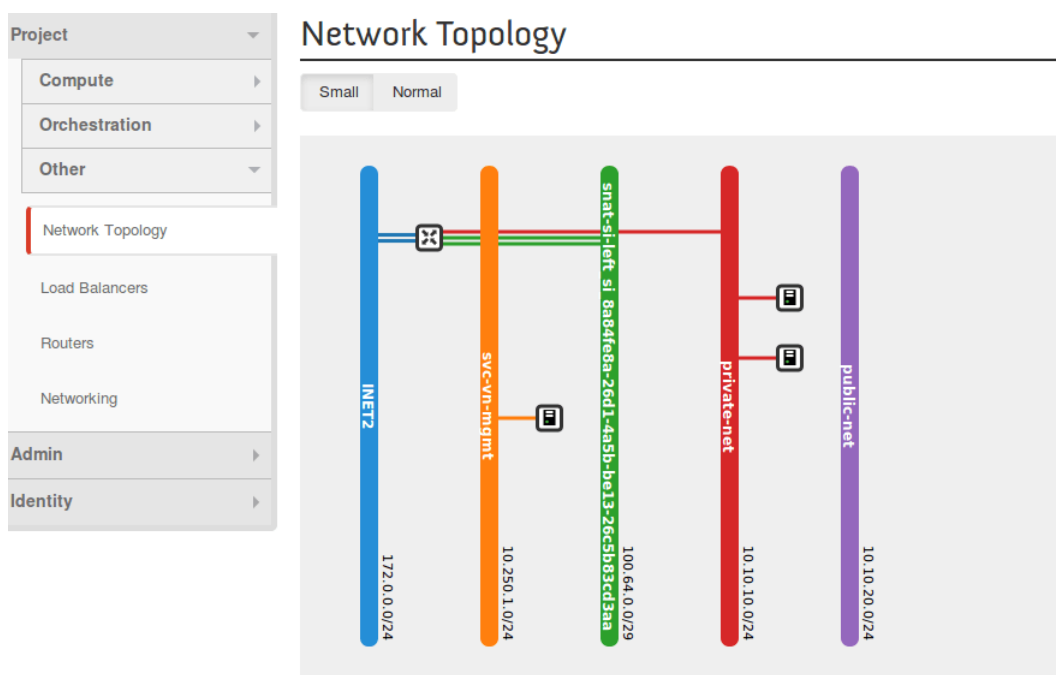
```
heat stack-create -f heat/templates/fwaas_mnmg_template.hot -e heat/env/fwaas_fortios_contrail.env fortios
```

By default, pfsense firewall is configured to NAT after the heat stack is started. As a result, there is no need to make any configuration for this function. Pfsense image was preconfigured with DHCP services on every interface and there is outbound policy for NAT.

After we start the heat with pfsense there is already functional service chaining. Testing instance has default gateway to contrail and contrail redirects it to pfsense.



Obrázek 4.4: Vytvořený health monitor



Obrázek 4.5: Vytvořená síťová topologie

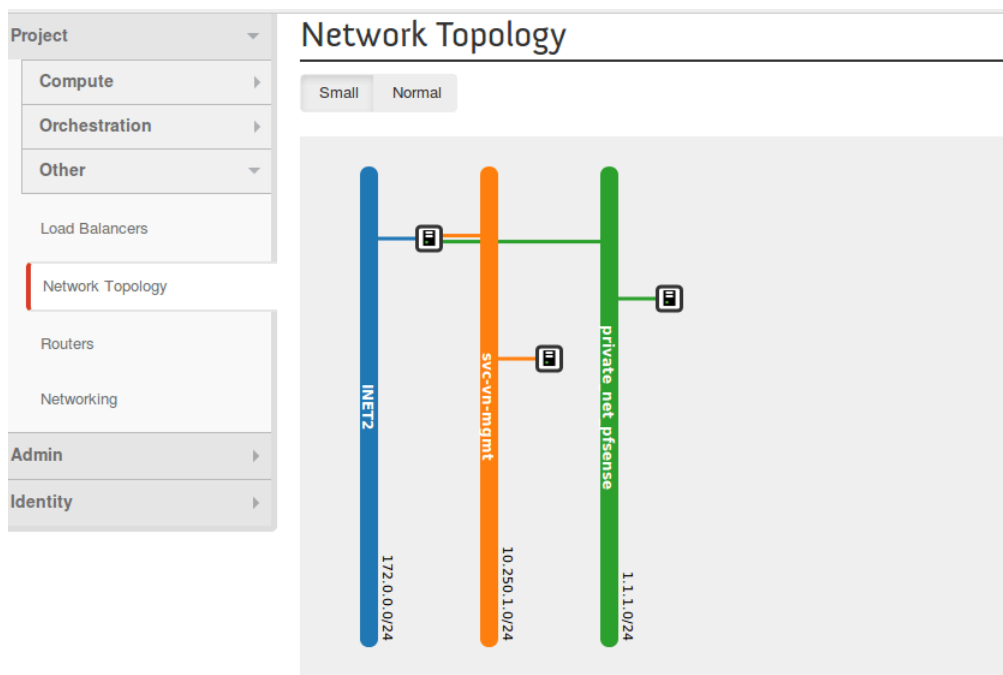
There is also NAT session in pfsense. In shell run command:

4.4.1 Fortigate VM

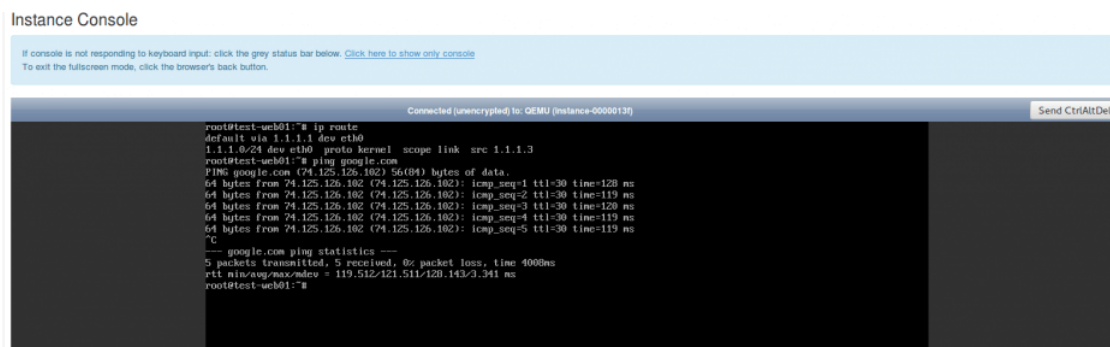
4.4.2 PFSense

```
File Edit View Search Terminal Help
root@Management:~# curl 172.0.0.6
Instance 01
root@Management:~# curl 172.0.0.6
Instance 01
root@Management:~# curl 172.0.0.6
Instance 02
root@Management:~# curl 172.0.0.6
Instance 02
root@Management:~# curl 172.0.0.6
Instance 01
root@Management:~# curl 172.0.0.6
Instance 01
root@Management:~# curl 172.0.0.6
Instance 02
root@Management:~# curl 172.0.0.6
Instance 02
root@Management:~# curl 172.0.0.6
Instance 01
root@Management:~# curl 172.0.0.6
Instance 01
root@Management:~# curl 172.0.0.6
Instance 02
root@Management:~# curl 172.0.0.6
Instance 02
root@Management:~# curl 172.0.0.6
Instance 01
root@Management:~# curl 172.0.0.6
Instance 01
root@Management:~# curl 172.0.0.6
Instance 02
root@Management:~# curl 172.0.0.6
Instance 02
root@Management:~# curl 172.0.0.6
Instance 01
root@Management:~#
```

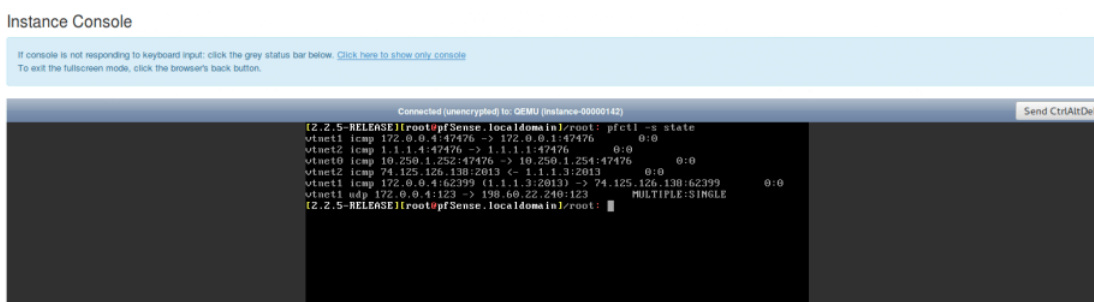
Obrázek 4.6: Test konektivity a load balancingu



Obrázek 4.7: Síťová topologie



Obrázek 4.8: Test konektivity PFSense



Obrázek 4.9: Ukázka NAT session

```

root@mnm01:~# python fortios_intf.py
This is the diff of the configs:

This is how to reach the desired state:
config system interface
    edit port1
        set allowaccess ssh ping http https
    next
    edit port2
        set defaultgw enable
    next
    edit port4
        set mode static
    next
    edit port5
        set mode static
    next
    edit port6
        set mode static
    next
    edit port7
        set mode static
    next
    edit ssl.root
        set mode static
    next
end
root@mnm01:~#

```

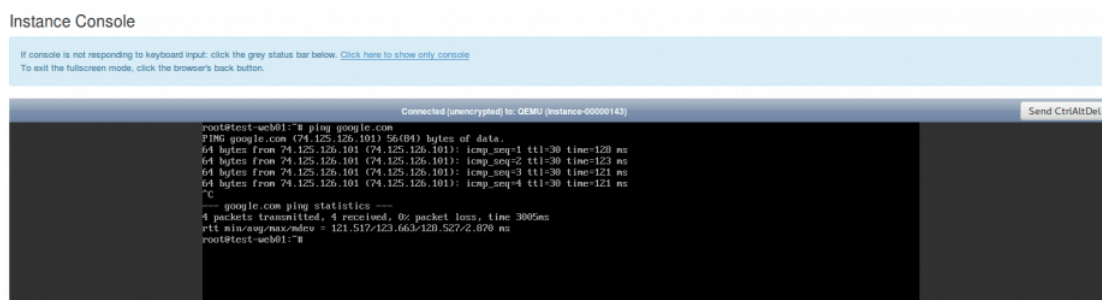
Obrázek 4.10: Fortigate VM intergace konfigurace

```
ubuntu@Management:~$ ssh root@172.0.0.5
Welcome to Ubuntu 14.04.3 LTS (GNU/Linux 3.19.0-26-generic x86_64)

 * Documentation:  https://help.ubuntu.com/
Last login: Tue Jan 12 10:03:49 2016 from mgmtserver14041vag
root@mnm01:~# ls
fabfile.py  fortigate-formula  fortios_intf.txt  fortios_nat.py  param.py  update.sh
fabfile.pyc  fortios_intf.py  fortios_nat.conf  fortios_nat.txt  text.py
root@mnm01:~# python fortios_nat.py
This is the diff of the conigs:

This is how to reach the desired state:
  config firewall policy
    edit 1
      set nat enable
      set service ALL
      set schedule always
      set srcaddr all
      set dstintf port2
      set srcintf port3
      set action accept
      set dstaddr all
      set logtraffic all
    next
  end
root@mnm01:~#
```

Obrázek 4.11: Fortigate VM NAT konfigurace



Obrázek 4.12: Test konektivity

5 Shrnutí poznatků

K čemu to je dobrý, na co jsem narazil, atd.

6 Závěr

Je v paráda.

Literatura

- [1] R. Guerzoni, "Network Functions Virtualisation: An Introduction, Benefits, Enablers, Challenges and Call for Action. Introductory white paper," in SDN and OpenFlow World Congress, June 2012. [online]. [cit. 2016-04-07]. Dostupné také z: https://portal.etsi.org/NFV/NFV_White_Paper.pdf
- [2]
- [3] MIJUMBI, Rashid, Joan SERRAT, Juan-Luis GORRICHIO, Niels BOUTEN, Filip DE TURCK a Raouf BOUTABA. *Network Function Virtualization: State-of-the-Art and Research Challenges*. IEEE Communications Surveys. 2016, 18(1), 236-262. DOI: 10.1109/COMST.2015.2477041. ISSN 1553-877x. Dostupné také z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7243304>
- [4] HAN, Bo, Vijay GOPALAKRISHNAN, Lusheng JI a Seungjoon LEE. *Network function virtualization: Challenges and opportunities for innovations*. IEEE Communications Magazine. 2015, 53(2), 90-97. DOI: 10.1109/MCOM.2015.7045396. ISSN 0163-6804. Dostupné také z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7045396>
- [5] KUSNETZKY, Dan. *Virtualization: a manager's guide*. Sebastopol, CA: O'Reilly, c2011. ISBN 1449306454.
- [6] J. Smith and R. Nair, "The architecture of virtual machines," Computer, vol. 38, no. 5, pp. 32-38, May 2005. doi: 10.1109/MC.2005.173
- [7] <http://www.rightscale.com/blog/cloud-industry-insights/cloud-computing-trends-2015-state-cloud-survey>
- [8] K. Chandrasekaran. *Essentials of CLOUD COMPUTING*. Boca Raton: CRC Press, 2015. ISBN 978-1-4822-0544-2.
- [9] JENNINGS, Brendan a Rolf STADLER. *Resource Management in Clouds: Survey and Research Challenges*. Journal of Network and Systems Management. 2015, 23(3), 567-619. DOI: 10.1007/s10922-014-9307-7. ISSN 1064-7570. Dostupné také z: <http://link.springer.com/10.1007/s10922-014-9307-7>

-
- [10] ETSI, “Network Function Virtualization: Use Cases”, http://www.etsi.org/deliver/etsi_gs/NFV/001_099/001/01.01.01_60/gs_NFV001v010101p.pdf, 2013
- [11] KREUTZ, Diego, Fernando M. V. RAMOS, Paulo ESTEVES VERISSIMO, Christian ESTEVE ROTHENBERG, Siamak AZODOLMOLKY a Steve UHLIG. *Software-Defined Networking: A Comprehensive Survey*. Proceedings of the IEEE [online]. 2015, 103(1), 14-76 [cit. 2016-04-09]. DOI: 10.1109/JPROC.2014.2371999. ISSN 0018-9219. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6994333>
- [12] DOHERTY, Jimmy. *SDN and NFV simplified: a visual guide to understanding software defined networks and network function virtualization*. 1st edition. Indianapolis, IN: Addison-Wesley Professional, 2016. ISBN 9780134306407.
- [13] ETSI Industry Specification Group (ISG) NFV, “ETSI GS NFV 002 V1.2.1: Network Functions Virtualisation (NFV); Architectural Framework,” December 2014. [online]. [cit. 2016-04-07]. Dostupné také z: <http://www.etsi.org/deliver/etsigs/NFV/001099/002/01.02.0160/gsNFV002v010201p.pdf>
- [14] ETSI Industry Specification Group (ISG) NFV, “ETSI GS NFV 003 V1.2.1: Network Functions Virtualisation (NFV); Terminology for Main Concepts in NFV,” December 2014. [online]. [cit. 2016-04-07]. <http://www.etsi.org/deliver/etsigs/NFV/001099/003/01.02.0160/gsNFV003v010201p.pdf>
- [15] http://www.etsi.org/deliver/etsi_gs/NFV-INF/001_099/001/01.01.01_60/gs_nfv-inf001v010101p.pdf
- [16] http://www.etsi.org/deliver/etsi_gs/NFV-INF/001_099/003/01.01.01_60/gs_NFV-INF003v010101p.pdf
- [17] http://www.etsi.org/deliver/etsi_gs/NFV-INF/001_099/004/01.01.01_60/gs_nfv-inf004v010101p.pdf
- [18] http://www.etsi.org/deliver/etsi_gs/NFV-INF/001_099/005/01.01.01_60/gs_NFV-INF005v010101p.pdf
- [19] http://www.etsi.org/deliver/etsi_gs/NFV-SWA/001_099/001/01.01.01_60/gs_nfv-swa001v010101p.pdf
- [20] http://www.etsi.org/deliver/etsi_gs/NFV-MAN/001_099/001/01.01.01_60/gs_nfv-man001v010101p.pdf

- [21] MIJUMBI, Rashid, Joan SERRAT, Juan-luis GORRICHIO, Steven LATRE, Mari-nos CHARALAMBIDES a Diego LOPEZ. *Management and orchestration challenges in network functions virtualization*. IEEE Communications Magazine. 2016, 54(1), 98-105. DOI: 10.1109/MCOM.2016.7378433. ISSN 0163-6804. Dostupné také z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7378433>
<http://network-functions-virtualization.com/mano.html>
<http://www.alticelabs.com/content/WP-An-NFV-SDN-Enabled-Service-Provider.pdf>
<http://www.tmcnet.com/tmc/whitepapers/documents/whitepapers/2013/9377-network-functions-virtualization-challenges-solutions.pdf>
<http://link.springer.com/article/10.1186/s13638-015-0450-y>
<http://link.springer.com/article/10.1007/s11036-015-0630-3>
- [22] Open platform for nfv. <https://www.opnfv.org/>. Accessed September 28, 2014.

Přílohy

Seznam obrázků

2.1	Koncept virtualizace síťových funkcí (NFV)	4
2.2	Schéma hypervisorů	5
2.3	Schéma SDN, převzato z [11]	8
2.4	NFV architektura, převzato z [13]	10
2.5	Schéma NFV infrastruktury	11
2.6	Schéma virtuální síťové funkce	12
2.7	Ukázka klasického service chainu pomocí fyzických síťových prvků . .	13
2.8	Ukázka VNF service chainu	13
2.9	Schéma NFV MANO	14
3.1	Architektura NFV řešení	16
3.2	Popis heat orchestrace	17
4.1	Testovací topologie	20
4.2	Vytvořený pool	21
4.3	Vytvoření members	21
4.4	Vytvořený health monitor	22
4.5	Vytvořená síťová topologie	22
4.6	Test konektivity a load balancingu	23
4.7	Síťová topologie	23
4.8	Test konektivity PFSense	24
4.9	Ukázka NAT session	24
4.10	Fortigate VM intergace konfigurace	24
4.11	Fortigate VM NAT konfigurace	25
4.12	Test konektivity	25

Seznam tabulek

Seznam ukázek kódu

Podklad pro zadání DIPLOMOVÉ práce studenta

PŘEDKLÁDÁ:	ADRESA	OSOBNÍ ČÍSLO
Smola Ondřej	Polizy 16, Osice - Polizy	11475

TÉMA ČESKY:

Orchestrace a management virtuálních síťových funkcí

TÉMA ANGLICKY:

Orchestration and management of virtual network functions

VEDOUcí PRÁCE:

Ing. Vladimír Soběslav, Ph.D. - KIT

ZÁSADY PRO VYPRACOVÁNÍ:

Cílem této práce je analyzovat možnosti vytváření a nasazení virtuálních sítí v cloud computingu s důrazem na technologie VNF nad NFV a jejich srovnání. V rámci závěrečné práce budou analyzovány metody a nástroje pro vývoj a automatizaci služeb virtuálních sítí. V závěrečné části provede autor implementaci VNF řešení v prostředí cloud computingové platformy OpenStack.

Osnova:

1. Úvod
2. Problematika virtualizace síťových funkcí
3. Testovací prostředí
4. Příklad virtualizace síťových funkcí
5. Shrnutí
6. Závěr

SEZNAM DOPORUČENÉ LITERATURY:

DOSTÁLEK, Libor.; KABELOVÁ, Alena. Velký průvodce protokoly TCP/IP a systémem DNS. 5. aktualizované vydání, Brno: Computer Press, a.s., 2008. 488 s. ISBN 978-80-251-2236-5.

HICKS, Michael. Optimizing Applications on Cisco Networks. 1. vydání. Indianapolis: Cisco Press, 2004. 384 s. ISBN: 978-1-58705-153-1.

HUCABY, David. CCNP SWITCH 642-813 Official Certification Guide. 1. vydání. Indianapolis: Cisco Press, 2011, 533 s. ISBN 978-1-58720-243-8.

Podpis studenta:

Datum:

Podpis vedoucího práce:

Datum: