

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 9. 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

<b>1 Úvod</b>	<b>1</b>
<b>2 Základní problematika virtualizace síťových funkcí</b>	<b>3</b>
2.1 Příklady využití . . . . .	5
2.2 NFV a Cloud Computing . . . . .	5
2.3 NFV a SDN . . . . .	5
2.4 Architektura NFV a VNF . . . . .	5
2.4.1 Infrastruktura NFV . . . . .	6
2.4.2 Virtuální síťová funkce . . . . .	7
2.4.3 Management a orchestrace NFV . . . . .	8
<b>3 Popis navrženého řešení a použitých technologií</b>	<b>9</b>
3.1 OpenStack . . . . .	9
3.1.1 Heat Templates . . . . .	9
3.2 OpenContrail . . . . .	9
3.2.1 Service Chaining . . . . .	10
<b>4 Testování navrženého řešení</b>	<b>11</b>
4.1 Testovací topologie . . . . .	11
4.2 Testované síťové funkce . . . . .	11
4.3 Heat template pro LbaaS . . . . .	12
4.3.1 Testování LbaaS . . . . .	13
4.4 Heat template pro FwaaS . . . . .	14
4.4.1 Testování FwaaS . . . . .	15
<b>5 Shrnutí poznatků</b>	<b>19</b>
<b>6 Závěr</b>	<b>20</b>
<b>Literatura</b>	<b>21</b>
<b>Přílohy</b>	<b>I</b>

# 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 vyžívána 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 popis 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.

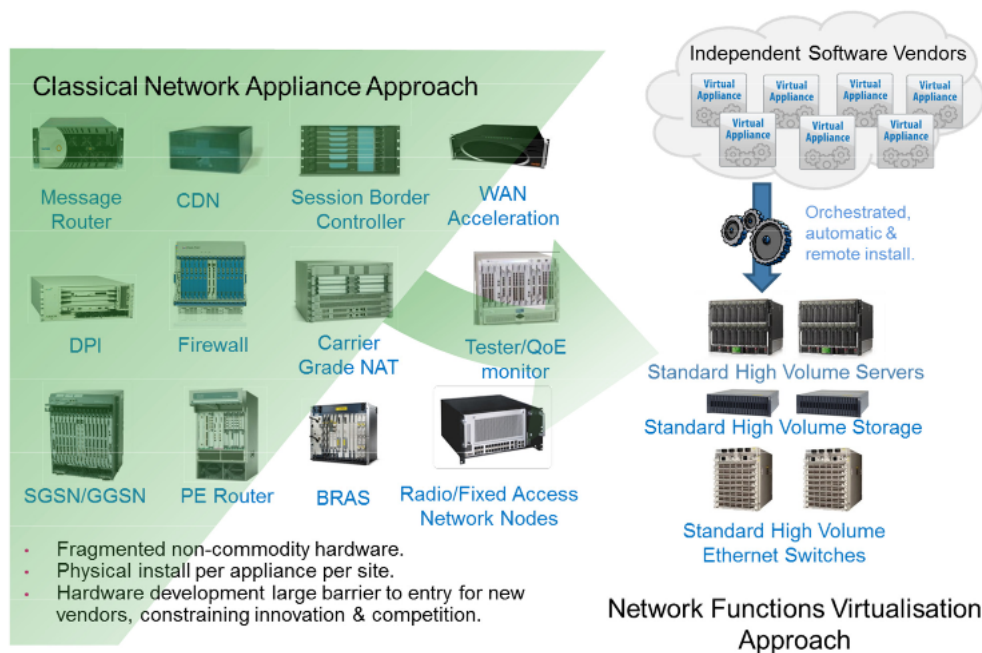
Pro vyřešení toho problému bylo navrženo v publikacích [1] a [2] skupinou několika telekomunikačních provozovatelů řešení ve formě virtualizace síťových funkcí (network functions virtualization). Toto řešení má za cíl zlepšit následující aspekty provozu telekomunikačních sítí:

- 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

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. Hlavní koncept virtualizace síťových funkcí znázorňuje obrázek č. 2.1.



**Obrázek 2.1:** Koncept virtualizace síťových funkcí (NFV)

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

- 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.

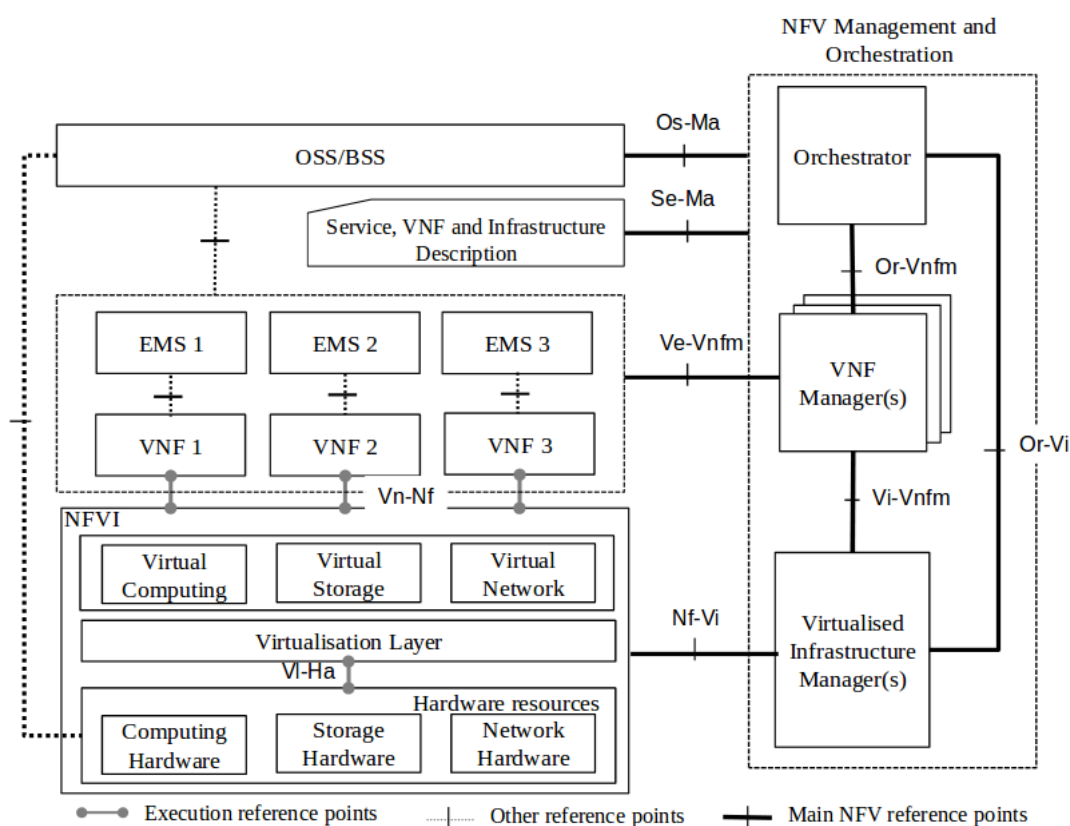
## 2.1 Příklady využití

## 2.2 NFV a Cloud Computing

## 2.3 NFV a SDN

## 2.4 Architektura NFV a VNF

V předchozí sekci byla popsána myšlenka a motivace související s virtualizací síťových funkcí. Pro úspěšné splnění těchto záměrů byla v [3] navržena referenční architektura pro framework virtualizace síťových funkcí (NFV framework). !!! Jedná se pouze o funkční návrh bez náznaků konkrétní implementace.



Obrázek 2.2: NFV architektura, převzato z [3]

Jak je vidět na obrázku č. 2.2, 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 jednot-

livé VNF být nasazeny. Tato infrastruktura může být velice rozsáhlá, proto je její součástí i síť poskytující konektivitu mezi vzdálenými lokacemi infrastruktury.[4]

- 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. [4]

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.

### 2.4.1 Infrastruktura NFV

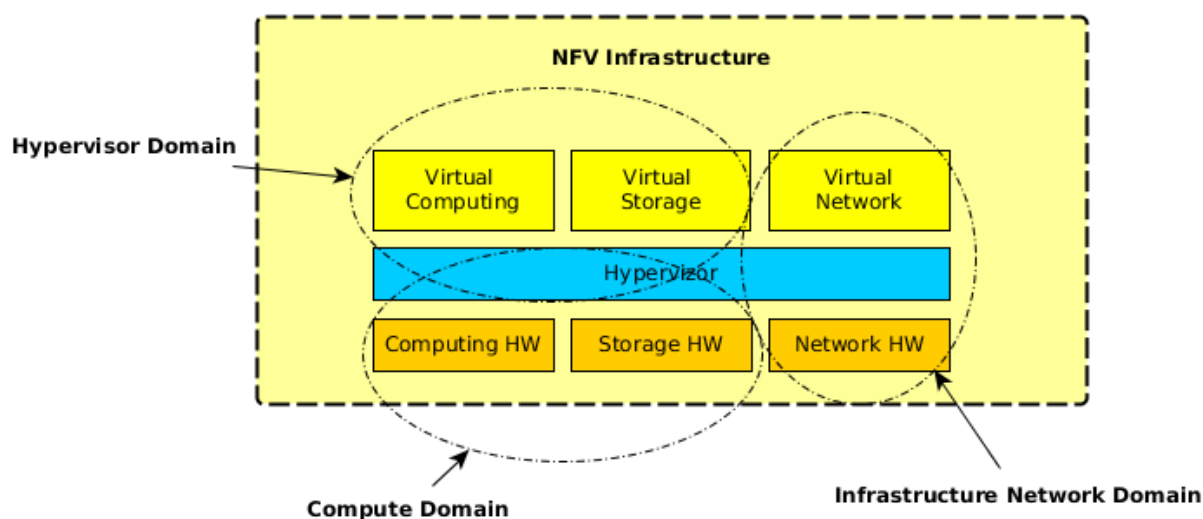
Ve zdroji [6], který detailně popisuje infrastrukturu pro virtualizaci síťových funkcí (NFVI), je uvedeno, že je v ní sdružení všech základních zdrojů potřebných pro běh virtuálních síťových funkcí (VNF). Z tohoto důvodu sem patří veškerý hardware. Do NFVI také patří některé softwarové komponenty, které jsou společné mnoha 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. [7]
- 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. [8]
- Infrastructure Network Domain - V této doméně je řešeno veškeré propojení výše zmíněných domén. Tedy fyzické i virtuální infrastruktury.[9]

Funkci obsaženou v jednotlivých doménách znázorňuje obrázek č. 2.3. Detailní popis těchto domén je nad rámec této práce. Více informací na tuto problematiku lze nalézt v [6] a ve zdrojích uvedených u každé domény. Dá se však říci, že návrh infrastruktury pro NFV je podobný jako pro návrh infrastruktury pro cloud computing platformu. Měl by se tedy skládat z generických a komerčně vysoce dostupných serverů, na které

je nasazen jeden z dostupných hypervisorů jako je např. KVM, Hyper-V či VMware. Tyto servery by následně měli být zapojeny do switchu a tím by měla být zajištěna konektivita.



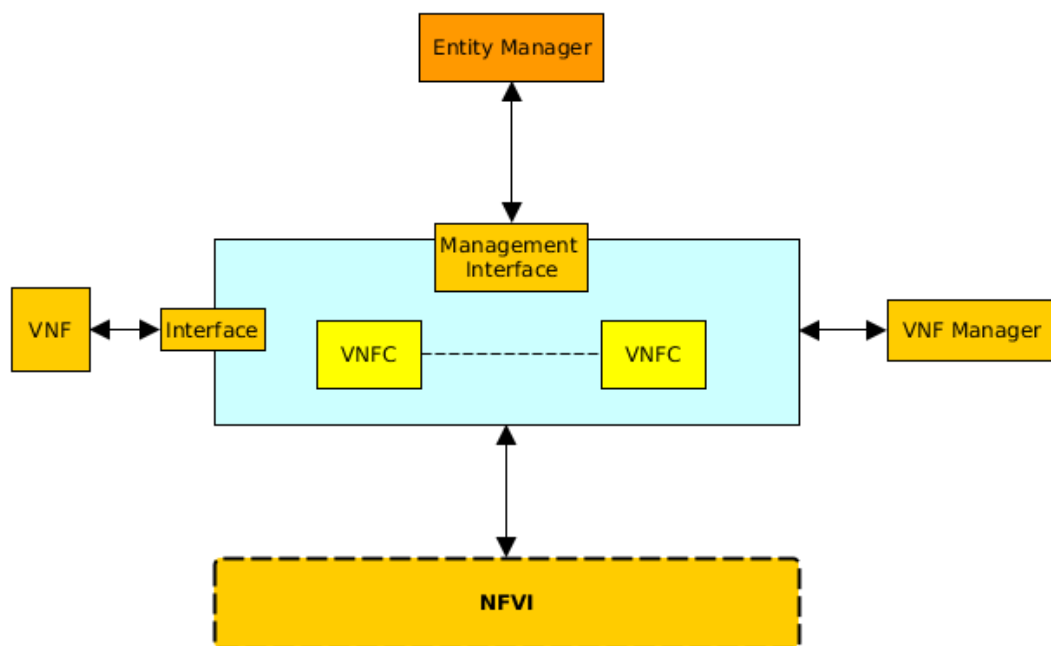
Obrázek 2.3: Schéma NFV infrastruktury

## 2.4.2 Virtuální síťová funkce

Virtuální síťová funkce (VNF) je dle [10] určitá síťová funkce, která běží na NFV infrastruktuře a je zároveň NFV frameworkem řízena a spravována. Zároveň musí mít dobře definovaný interface k ostatním síťovým funkcím, k VNF Managerovi a měla by obsahovat management interface.

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

Na obrázku č. 2.4 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 NFV 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 2.4: Schéma virtuální síťové funkce

## 2.4.3 Management a orchestrace NFV

### 2.4.3.1 Orchestrator

### 2.4.3.2 VNF manager

### 2.4.3.3 Virtualised Infrastructure Manager

## 3 Popis navrženého řešení a použitých technologií

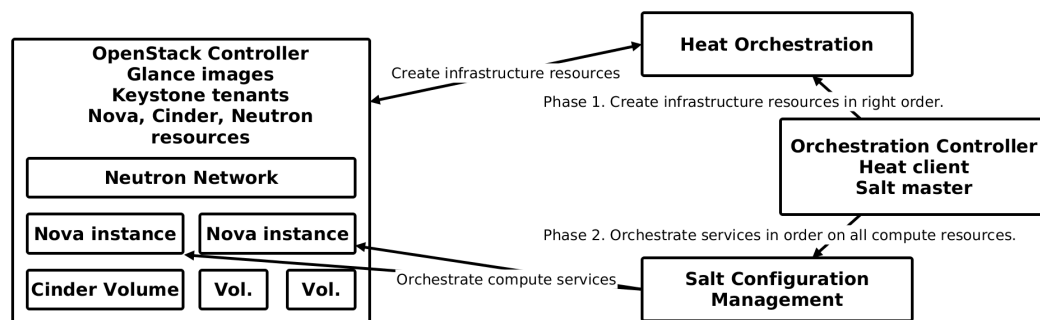
### 3.1 OpenStack

Popis Openstacku

#### 3.1.1 Heat Templates

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.



Obrázek 3.1: Popis heat orchestrace

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

### 3.2 OpenContrail

Popis OpenContrailu.

### **3.2.1 Service Chaining**

Popis service chaining v contrail a service instanci. a jak to může být využito pro VNF.

## 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.

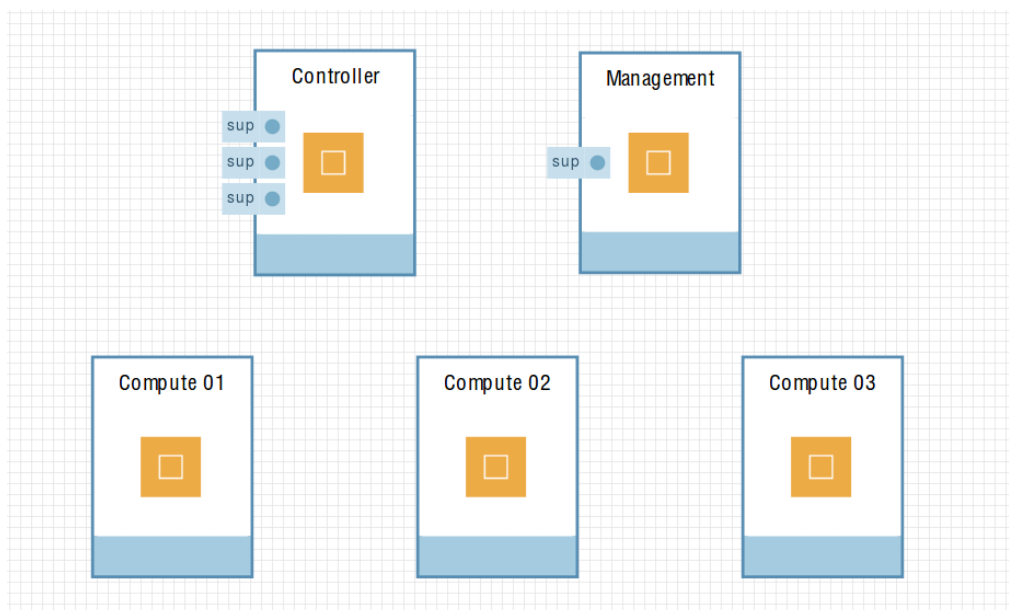
### 4.2 Testované síťové funkce

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

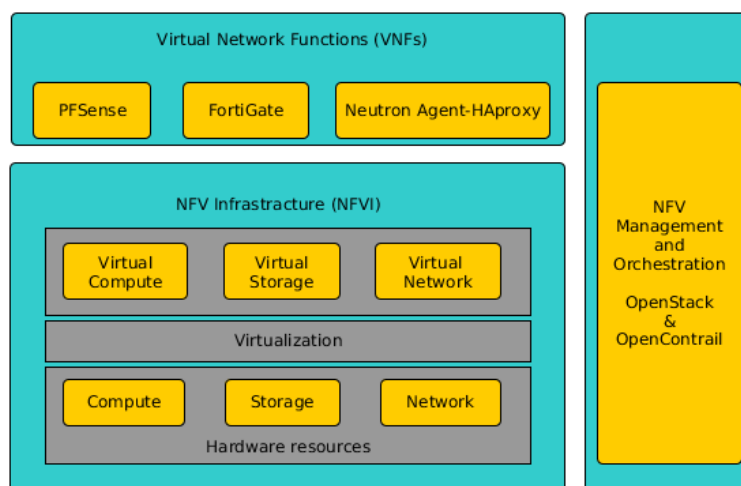
- Neutron Agent-HAproxy – je velmi rychlé a spolehlivé řešení nabízející vysokou dostupnost, load balancing a proxy pro aplikace založené na TCP a HTTP

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 Heat template pro LbaaS

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



Obrázek 4.2: Architektura NFV řešení

### 4.3.1 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/en-
nv/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.

Project

Compute

Orchestration

Other

Network Topology

Load Balancers

Routers

Networking

# Load Balancer

Pools

Members

Monitors

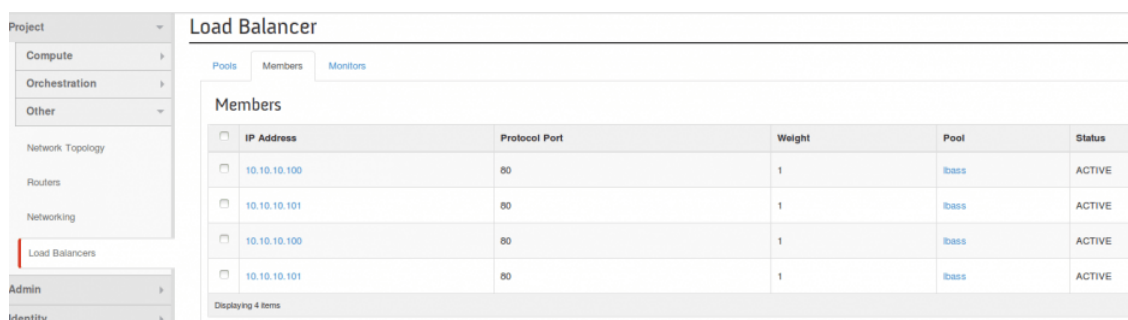
## Pools

<input type="checkbox"/>	Name	Description	Provider	Subnet	Protocol	Status	VIP
<input type="checkbox"/>	lbaas		opencontrail	10.10.10.0/24	HTTP	ACTIVE	lb_pool.vip

Displaying 1 item

Obrázek 4.3: Vytvořený pool

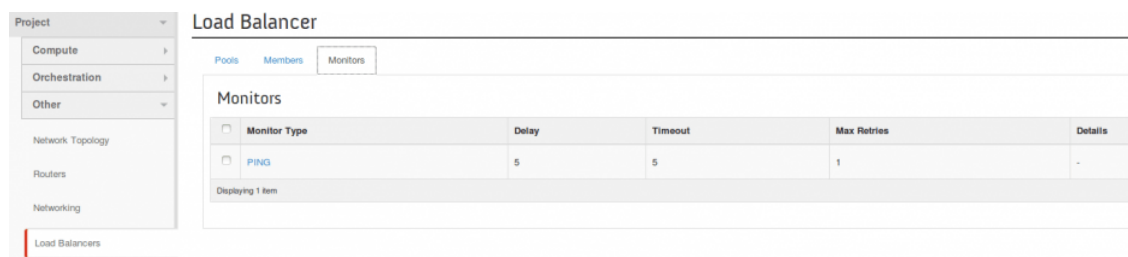
Další zdrojem, který byl vytvořen je health monitor, který lze vidět na obrázku č. X+2. Díky němu má load balancer přehled o aktuálním stavu webových instancí. Pokud by náhodou některá z nich přestala odpovídat, v tomto případě na ping, tak by load balancer na tuto instanci přestal zasílat traffic.



Load Balancer					
Members					
<input type="checkbox"/>	IP Address	Protocol Port	Weight	Pool	Status
<input type="checkbox"/>	10.10.10.100	80	1	lbass	ACTIVE
<input type="checkbox"/>	10.10.10.101	80	1	lbass	ACTIVE
<input type="checkbox"/>	10.10.10.100	80	1	lbass	ACTIVE
<input type="checkbox"/>	10.10.10.101	80	1	lbass	ACTIVE

Displaying 4 items

Obrázek 4.4: Vytvoření members



Load Balancer					
Monitors					
<input type="checkbox"/>	Monitor Type	Delay	Timeout	Max Retries	Details
<input type="checkbox"/>	PING	5	5	1	-

Displaying 1 item

Obrázek 4.5: Vytvořený health monitor

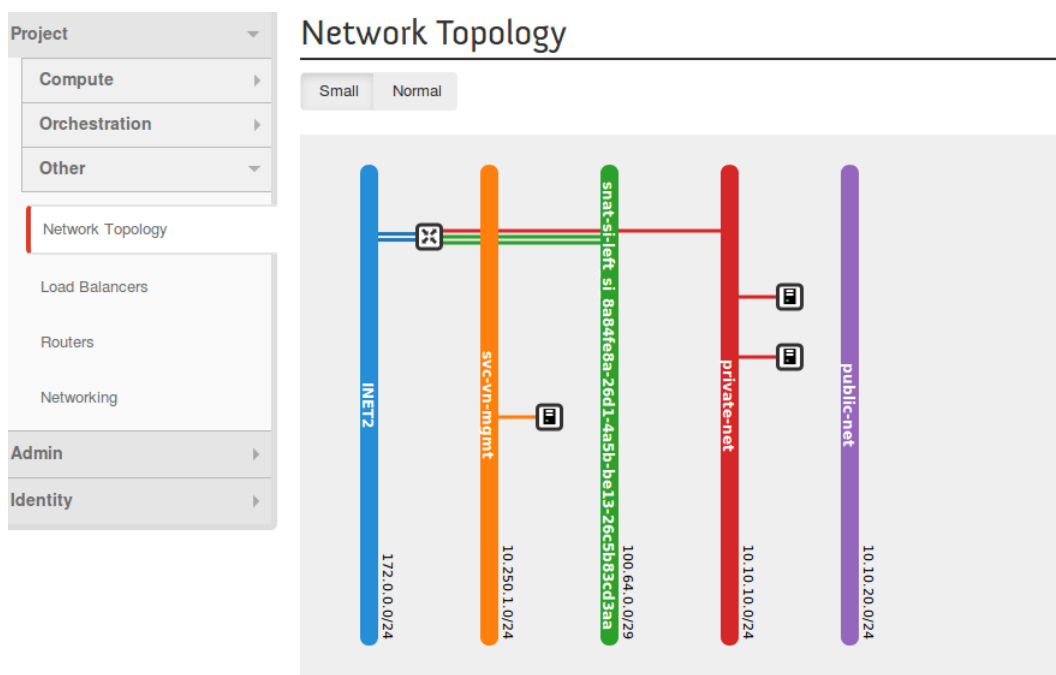
Finální síťovou topologií znázorňuje obrázek č. X+3.

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 Heat template pro Fwaas

Pro Fwaas je naruhot heat template, který obsahuje:

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



Obrázek 4.6: Vytvořená síťová topologie

#### 4.4.1 Testování Fwaas

Pro vytvoření heat stacku s PFSense z templaty 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
```

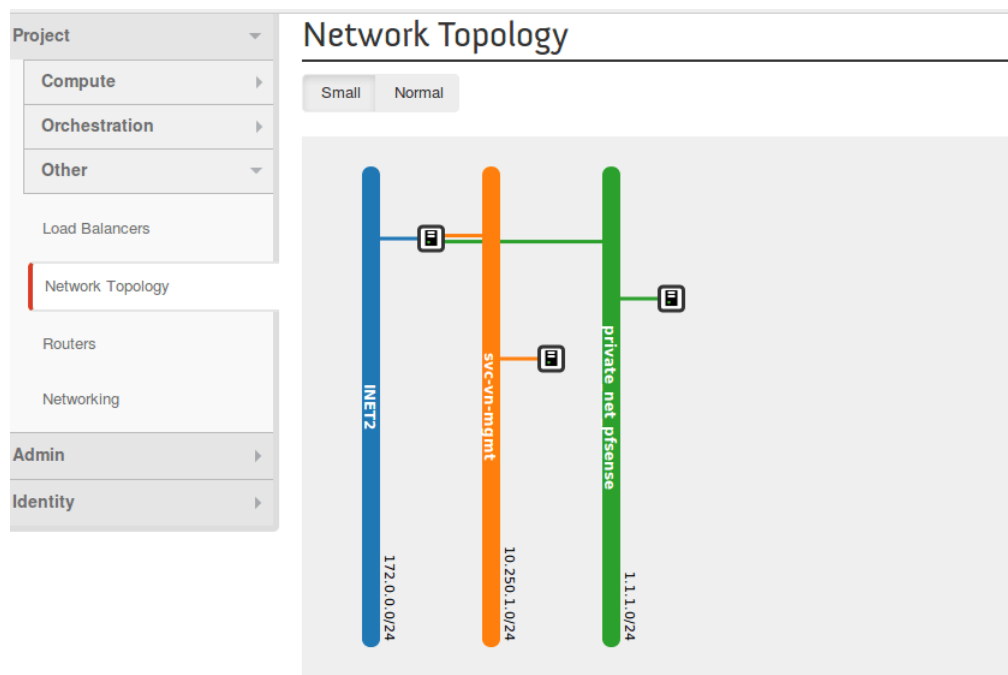
By default, pfsense firewall is configured to NAT after the heat stack is started. As a result, there is no need to make any configuration for this function. Pfsense image was preconfigured with DHCP services on every interface and there is outbound policy for NAT.

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.

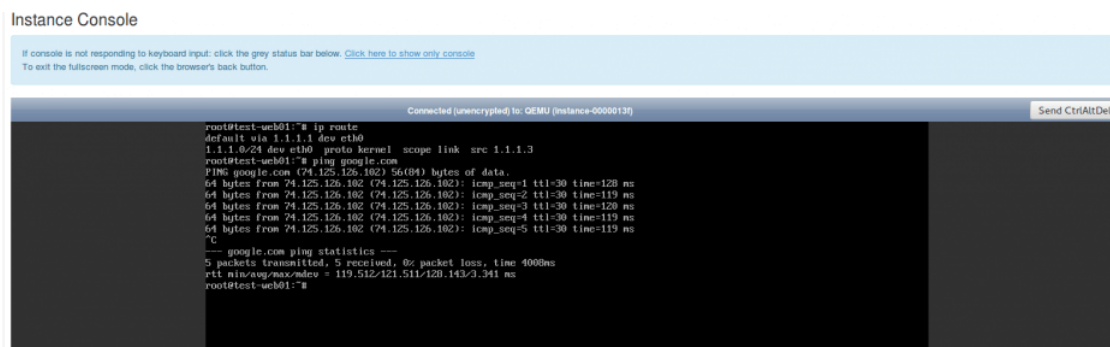
There is also NAT session in pfsense. In shell run command:

```
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:~#
```

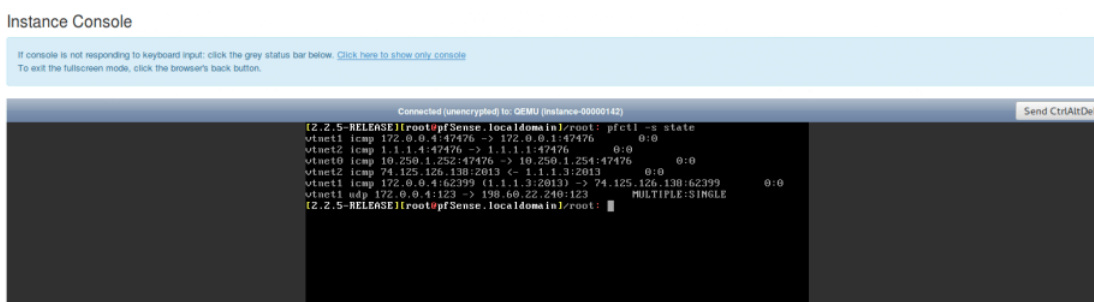
Obrázek 4.7: Test konektivity a load balancingu



Obrázek 4.8: Síťová topologie



Obrázek 4.9: Test konektivity PFSense



Obrázek 4.10: 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 4.11: Fortigate VM intergace konfigurace

```

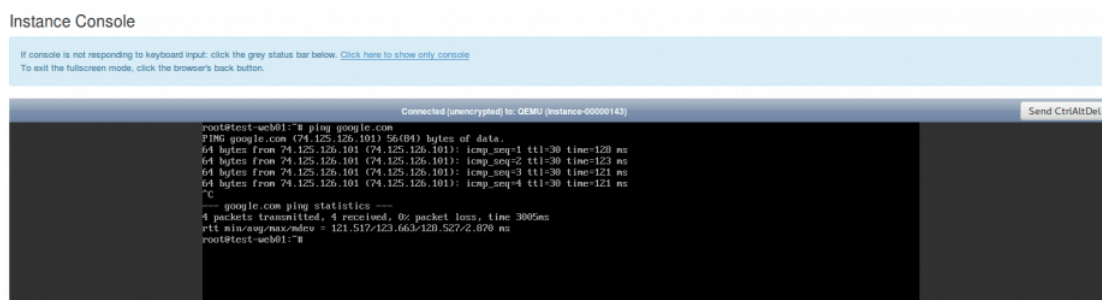
ubuntu@Management:~$ ssh root@172.0.0.5
Welcome to Ubuntu 14.04.3 LTS (GNU/Linux 3.19.0-26-generic x86_64)

 * Documentation:  https://help.ubuntu.com/
Last login: Tue Jan 12 10:03:49 2016 from mgmtserver14041vag
root@mnmg01:~# 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@mnmg01:~# 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@mnmg01:~# █

```

Obrázek 4.12: Fortigate VM NAT konfigurace



Obrázek 4.13: 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](https://portal.etsi.org/NFV/NFV_White_Paper.pdf)
- [2]
- [3] 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/etsisgs/NFV/001099/002/01.02.0160/gsNFV002v010201p.pdf>
- [4] 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/etsisgs/NFV/001099/003/01.02.0160/gsNFV003v010201p.pdf>
- [5] Open platform for nfv. <https://www.opnfv.org/>. Accessed September 28, 2014.
- [6] [http://www.etsi.org/deliver/etsi\\_gs/NFV-INF/001\\_099/001/01.01.01\\_60/gs\\_nfv-inf001v010101p.pdf](http://www.etsi.org/deliver/etsi_gs/NFV-INF/001_099/001/01.01.01_60/gs_nfv-inf001v010101p.pdf)
- [7] [http://www.etsi.org/deliver/etsi\\_gs/NFV-INF/001\\_099/003/01.01.01\\_60/gs\\_NFV-INF003v010101p.pdf](http://www.etsi.org/deliver/etsi_gs/NFV-INF/001_099/003/01.01.01_60/gs_NFV-INF003v010101p.pdf)
- [8] [http://www.etsi.org/deliver/etsi\\_gs/NFV-INF/001\\_099/004/01.01.01\\_60/gs\\_nfv-inf004v010101p.pdf](http://www.etsi.org/deliver/etsi_gs/NFV-INF/001_099/004/01.01.01_60/gs_nfv-inf004v010101p.pdf)
- [9] [http://www.etsi.org/deliver/etsi\\_gs/NFV-INF/001\\_099/005/01.01.01\\_60/gs\\_NFV-INF005v010101p.pdf](http://www.etsi.org/deliver/etsi_gs/NFV-INF/001_099/005/01.01.01_60/gs_NFV-INF005v010101p.pdf)
- [10] [http://www.etsi.org/deliver/etsi\\_gs/NFV-SWA/001\\_099/001/01.01.01\\_60/gs\\_nfv-swa001v010101p.pdf](http://www.etsi.org/deliver/etsi_gs/NFV-SWA/001_099/001/01.01.01_60/gs_nfv-swa001v010101p.pdf)
- [11] [http://www.etsi.org/deliver/etsi\\_gs/NFV-MAN/001\\_099/001/01.01.01\\_60/gs\\_nfv-man001v010101p.pdf](http://www.etsi.org/deliver/etsi_gs/NFV-MAN/001_099/001/01.01.01_60/gs_nfv-man001v010101p.pdf)

- [12]
- [13] 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
- [14] ETSI, "Network Function Virtualization: Use Cases", [http://www.etsi.org/deliver/etsi\\_gs/NFV/001\\_099/001/01.01.01\\_60/gs\\_NFV001v010101p.pdf](http://www.etsi.org/deliver/etsi_gs/NFV/001_099/001/01.01.01_60/gs_NFV001v010101p.pdf), 2013
- [15] OPPENHEIMER, Priscilla. *Top-down network design*. 3rd ed. Indianapolis, IN: Cisco Press, c2011, xxiv, 447 p. ISBN 15-872-0283-2.
- [16] 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>
- [17] 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>
- [18] MIJUMBI, Rashid, Joan SERRAT, Juan-luis GORRICHIO, Steven LATRE, Marinós 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>
- [19] 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>  
<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>

# **Přílohy**

# Seznam obrázků

2.1	Koncept virtualizace síťových funkcí (NFV) . . . . .	4
2.2	NFV architektura, převzato z [3] . . . . .	5
2.3	Schéma NFV infrastruktury . . . . .	7
2.4	Schéma virtuální síťové funkce . . . . .	8
3.1	Popis heat orchestrace . . . . .	9
4.1	Testovací topologie . . . . .	12
4.2	Architektura NFV řešení . . . . .	13
4.3	Vytvořený pool . . . . .	13
4.4	Vytvoření members . . . . .	14
4.5	Vytvořený health monitor . . . . .	14
4.6	Vytvořená síťová topologie . . . . .	15
4.7	Test konektivity a load balancingu . . . . .	16
4.8	Síťová topologie . . . . .	16
4.9	Test konektivity PFSense . . . . .	17
4.10	Ukázka NAT session . . . . .	17
4.11	Fortigate VM intergace konfigurace . . . . .	17
4.12	Fortigate VM NAT konfigurace . . . . .	18
4.13	Test konektivity . . . . .	18

## Seznam tabulek

## **Seznam ukázek kódu**



**Podklad pro zadání DIPLOMOVÉ práce studenta**

<b>PŘEDKLÁDÁ:</b>	<b>ADRESA</b>	<b>OSOBNÍ ČÍSLO</b>
Smola Ondřej	Polizy 16, Osice - Polizy	I1475

**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: .....