

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 3. 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</b>	<b>Úvod</b>	<b>1</b>
<b>2</b>	<b>Základní problematika virtualizace síťových funkcí</b>	<b>2</b>
2.1	Princip virtualizace . . . . .	2
2.2	Cloud computing . . . . .	3
2.2.1	Model nasazení . . . . .	3
2.2.2	Distribuční model . . . . .	3
2.2.3	Výhody . . . . .	4
2.3	Softwarově definované sítě . . . . .	4
2.4	Virtualizované síťové funkce . . . . .	4
2.4.1	Základní architektura . . . . .	5
2.4.2	Management a orchestrace . . . . .	5
<b>3</b>	<b>Popis použitých technologií a testovacího prostředí</b>	<b>6</b>
3.1	OpenStack . . . . .	6
3.1.1	Heat Templates . . . . .	6
3.2	OpenContrail . . . . .	6
3.2.1	Service Chaining . . . . .	7
3.3	Testovací topologie . . . . .	7
3.4	Testované síťové funkce . . . . .	7
<b>4</b>	<b>Návrh řešení virtualizace síťových funkcí</b>	<b>9</b>
4.1	Heat template pro LbaaS . . . . .	9
4.1.1	Testování LbaaS . . . . .	10
4.2	Heat template pro FwaaS . . . . .	12
4.2.1	Testování FwaaS . . . . .	12
<b>5</b>	<b>Shrnutí poznatků</b>	<b>16</b>
<b>6</b>	<b>Závěr</b>	<b>17</b>
	<b>Literatura</b>	<b>18</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 v oblasti výpočetního výkonu a úložiště. Je již běžnou praxí, že v datových centrech vše běží na jedné fyzické infrastruktuře, na které existuje několik různých projektů zcela oddělených virtuálním prostředím. K vývoji došlo i v oblasti počítačových sítí. Díky softwarově definovaným sítím je možné vytvářet na sobě nezávislé sítě a vytvářet tak různé síťové topologie.

Avšak i přes tyto nové technologie je dnes nejvíce síťové funkčnosti zatím 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. Tento fakt se snaží vyřešit právě virtualizované síťové funkce, na kterou se zaměřuje tato diplomová práce.

Celá struktura této práce je rozdělena na 3 hlavní části. První dvě části jsou popisují oblast virtualizace síťových funkcí z teoretického hlediska a poslední pak z hlediska praktického. V první kapitole jsou vysvětleny hlavní pojmy a problematika této oblasti. Druhá je věnována popisu použitých technologií OpenStack a OpenContrail. V třetí části je následně ukázáno několik praktických příkladů. 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

Tato kapitola se zabývá základní problematikou virtuálních síťových funkcí. Pro pochopení konceptu virtuálních síťových funkcí je nejprve nutné vysvětlení několika základních pojmů a oblastí se kterou se souvisí. Proto jsou v této kapitole postupně popsány základní principy virtualizace, cloud computingu, softwarově definovaných sítí a následně již samotná oblast virtualizace síťových funkcí.

### 2.1 Princip virtualizace

Virtualizace je hlavní technologie, která umožnila vývoj nových řešení používaných v moderních datových centrech po celém světě. Základní pojmy.

Virtualní stroj (VM)

Hypervizory

Host OS

Guest OS

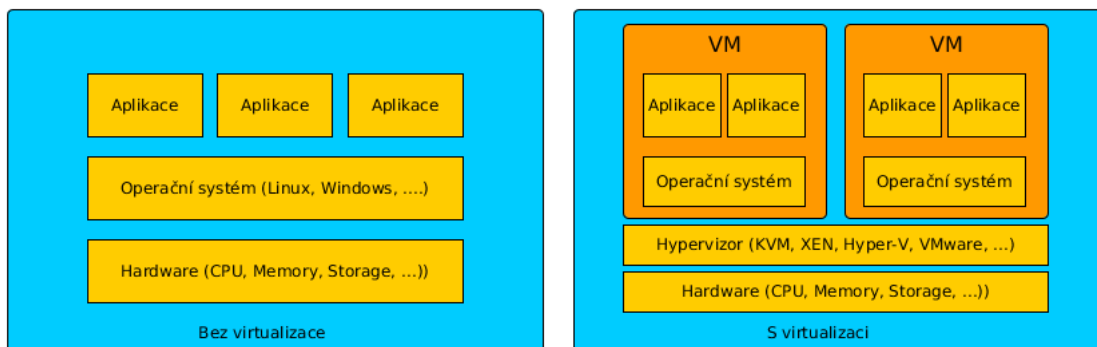
Přestože cíl virtualizace je vždy stejný, tak existuje více přístupů, jak ho dosáhnout.

V praxi se dnes používají tyto 2 hlavní metody virtualizace:

- Úplná virtualizace
- Částečná virtualizace
- Paravirtualizace

Typy hypervizorů:

- Typ 1 - nativní (Bare-metal)
- Typ 2 - hostovaný



Obrázek 2.1: Základní znázornění virtualizace

## 2.2 Cloud computing

Cloud Computing je hlavní oblast, která virtualizaci využívá. Z tohoto důvodu bude v této práci stručně vysvětlen.

Obrázek cloudu.

### 2.2.1 Model nasazení

Existuje několik modelů, které určují, jak může být cloud nasazen. Tyto modely jsou vždy závislé na uživatelských požadavcích a potřebách. Uživatel za základě těchto parametrů může vybírat z těchto čtyř možných modelů pro nasazení cloudu.

- Privátní cloud
- Veřejný cloud
- Hybridní cloud
- Komunitní cloud

Privátní cloud je cloudové prostředí vytvořené pro jednu organizaci.

Veřejný cloud je cloudová infrastruktura, která je k dispozici k užívání veřejnosti.

Hybridní cloud je

Posledním modelem nasazení je komunitní cloud. V tomto modelu je

### 2.2.2 Distribuční model

SaaS

PaaS

IaaS

NaaS



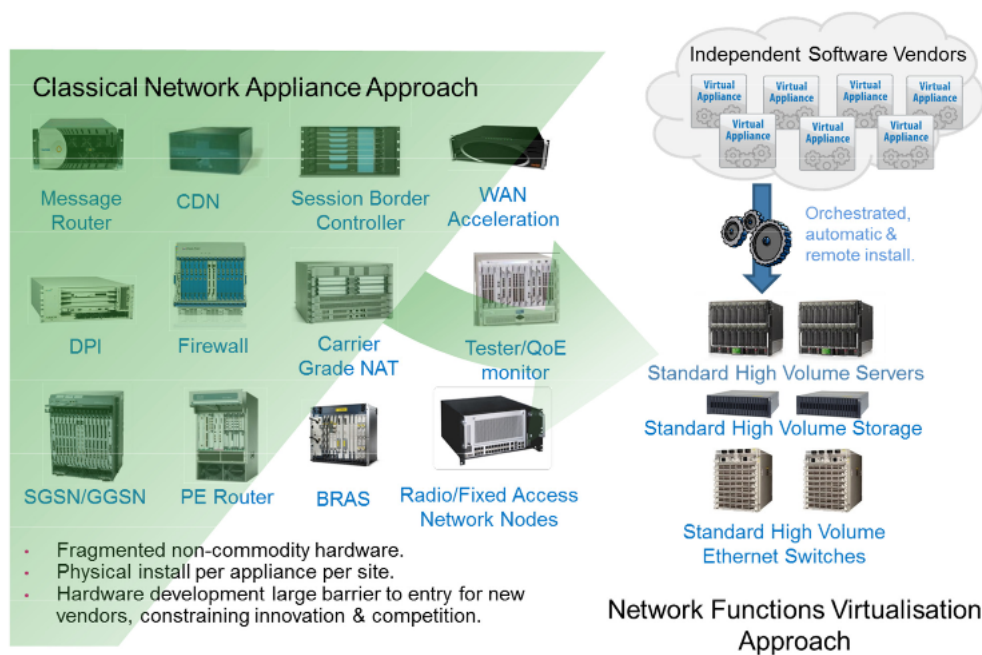
### 2.2.3 Výhody

## 2.3 Softwarově definované sítě

Architektura sítí - virtualizace síťové infrastruktury a návrh nových protokolů

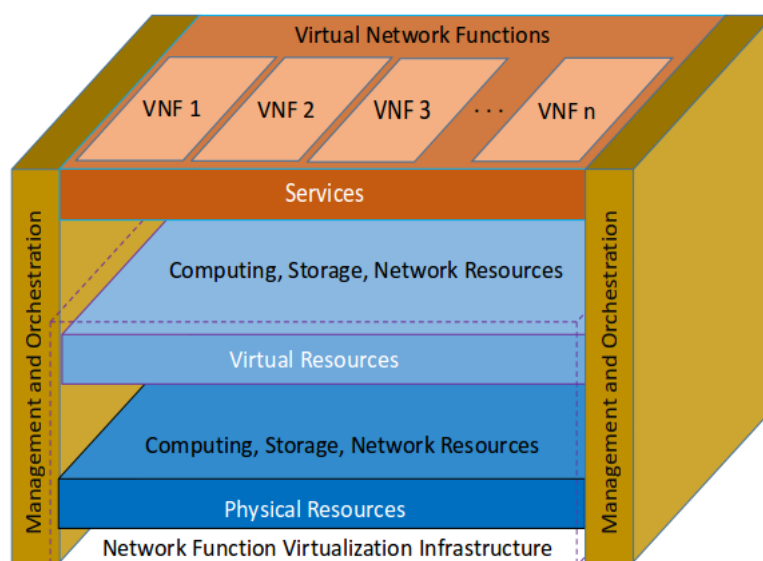
## 2.4 Virtualizované síťové funkce

- Router
- Firewall
- IPS
- IDS
- Load-balancer
- ...



Obrázek 2.2: Vize Virtualizace síťových funkcí

Tady bude napsaný rozdíl mezi tradičním HW přístupem a virtualizovaným přístupem



Obrázek 2.3: NFV architektura

#### 2.4.1 Základní architektura

#### 2.4.2 Management a orchestrace

Obrázek s popisem VNF frameworku  
MANO TOSCA a podobný

## 3 Popis použitých technologií a testovacího prostředí

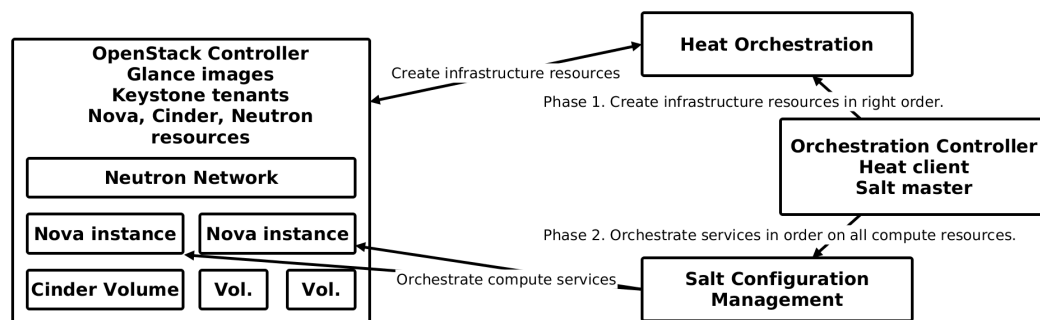
### 3.1 OpenStack

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.

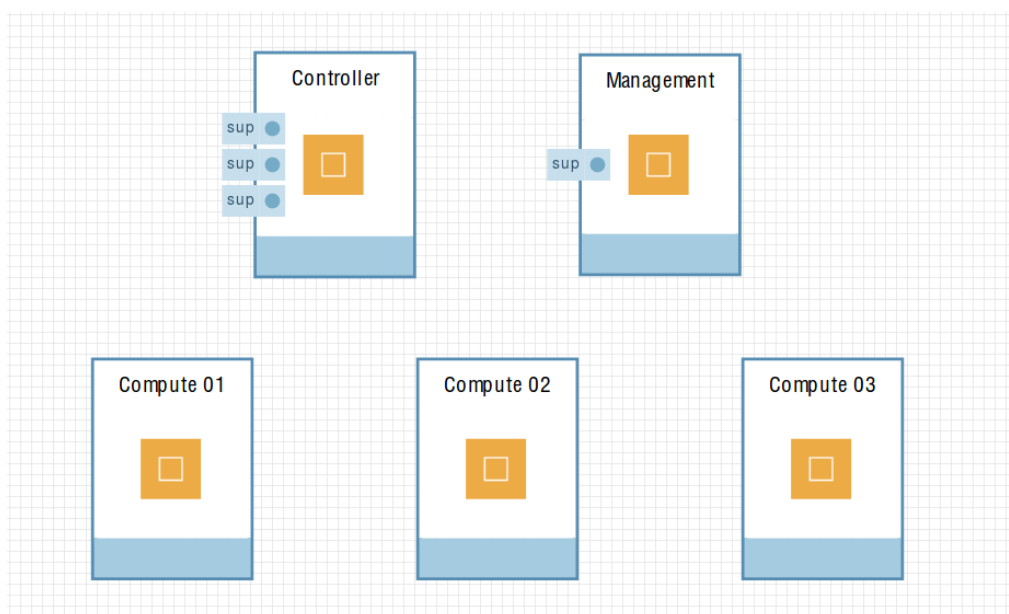
### 3.2.1 Service Chaining

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

## 3.3 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.



Obrázek 3.2: Testovací topologie

## 3.4 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 šablon, které se spouští v prostředí OpenStack.

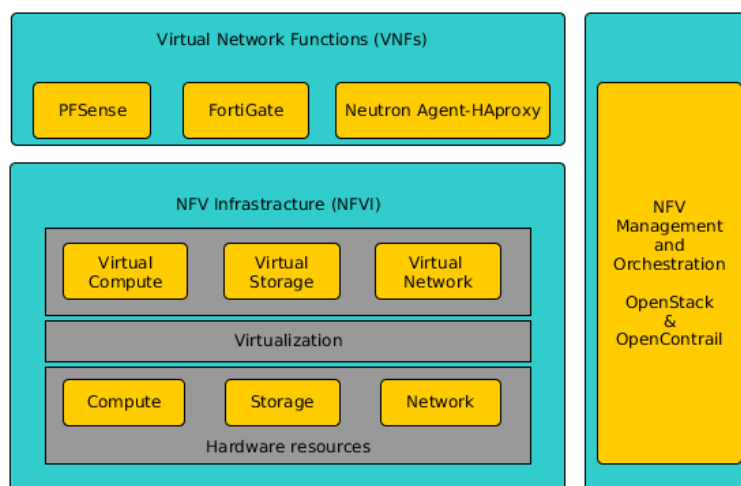
Aby mohla být nějaká VNF vůbec vytvořena, tak musel být nejprve zvolen software či operační systém, který má požadovanou funkci implementovanou. Pro tyto účely byly použity následující řešení:

- PFSense – open-source firewall založený na operačním systému FreeBSD.
- FortiGate-VM – je plnohodnotně vybavený Fortigate firewall zabalený jako virtuální instance.
- Neutron Agent-HAproxy – je velmi rychlé a spolehlivé řešení nabízející vysokou dostupnost, load balancing a proxy pro aplikace založené na TCP a HTTP

## 4 Návrh řešení virtualizace síťových funkcí

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.

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



Obrázek 4.1: Architektura NFV řešení

### 4.1 Heat template pro LbaaS

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

- pool

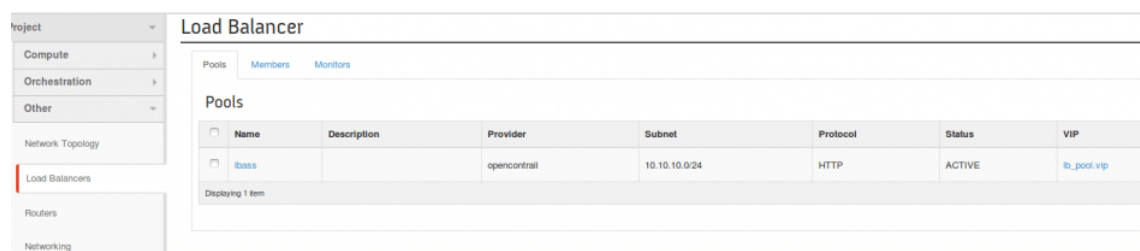
- members
- health monitoring
- 2 web instance
- privatní síť
- public síť

#### 4.1.1 Testování LbaaS

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

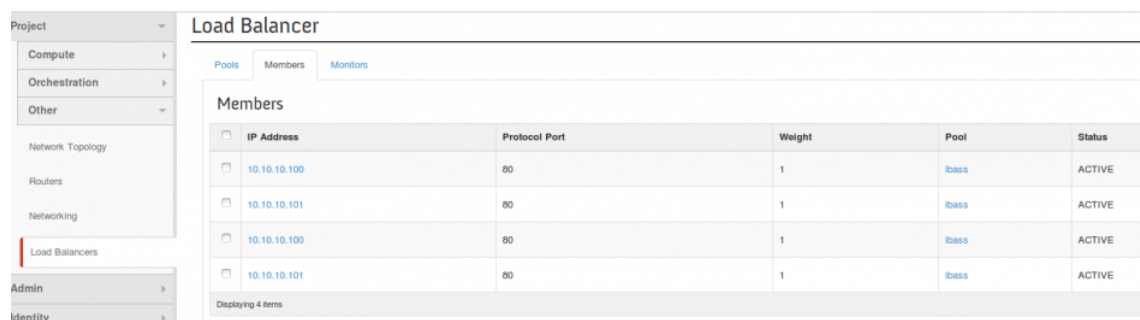
```
heat stack-create -f heat/templates/lbaas_template.hot -e heat/env/lbaas_env.env lbaas
```

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.



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

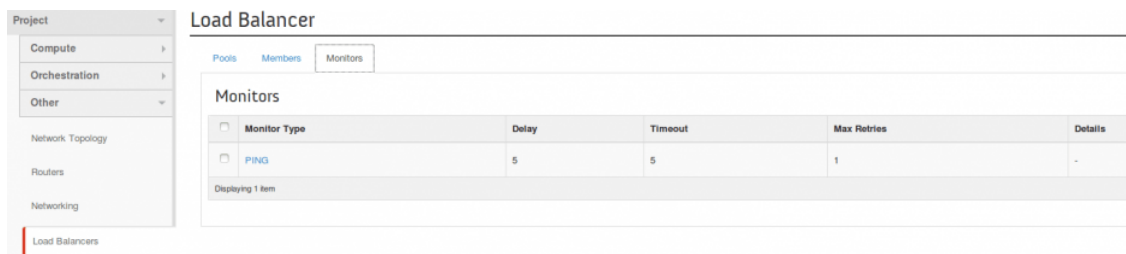
Obrázek 4.2: Vytvořený pool



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

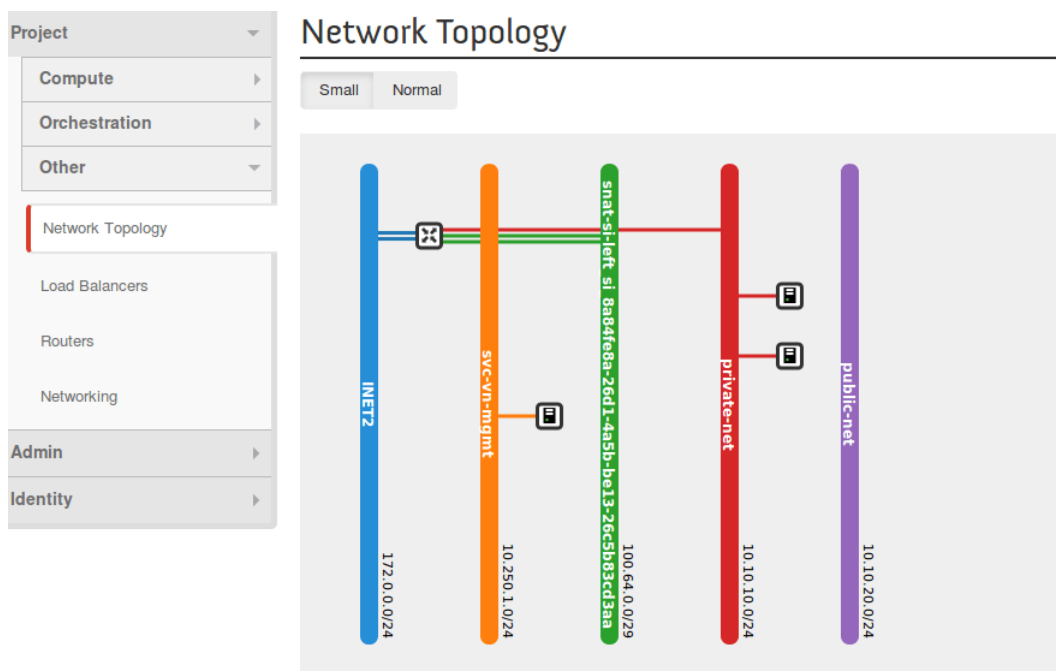
Obrázek 4.3: Vytvoření members

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.



Obrázek 4.4: Vytvořený health monitor

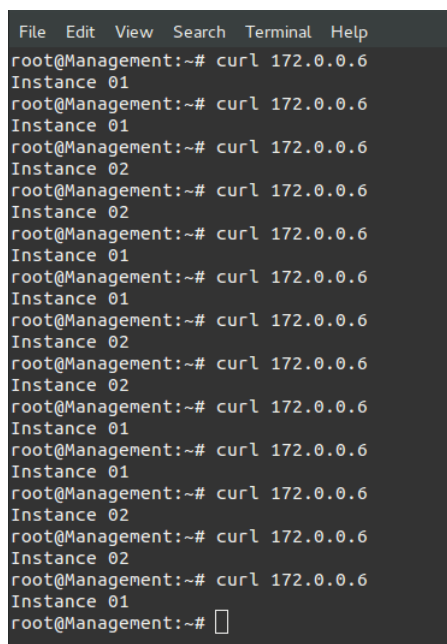
Finální síťovou topologii znázorňuje obrázek č. X+3.



Obrázek 4.5: Vytvořená síťová topologie

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





```

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:~# 

```

Obrázek 4.6: Test konektivity a load balancingu

## 4.2 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

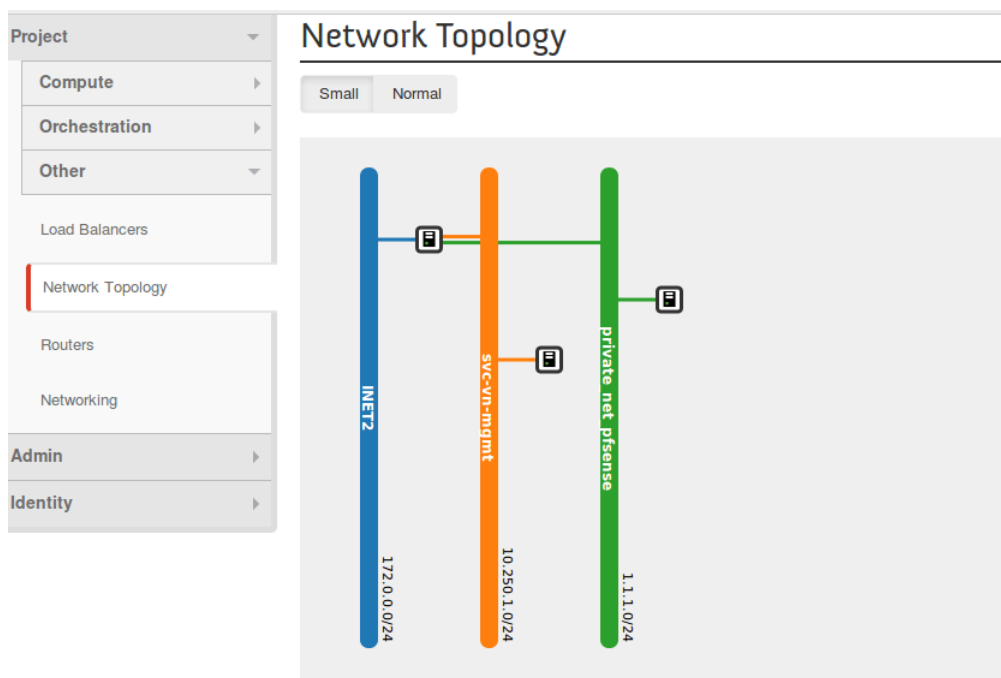
### 4.2.1 Testování FwaaS

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

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



Obrázek 4.7: Síťová topologie

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.

Instance Console

If console is not responding to keyboard input, click the grey status bar below. [Click here to show only console](#)  
To exit the fullscreen mode, click the browser's back button.

Connected (unencrypted) to: QEMU (instance-0000013f) Send CtrlAltDel

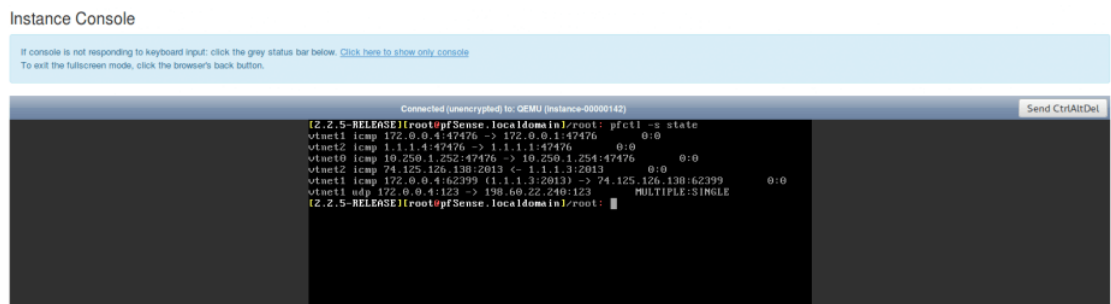
```

root@test-ueb01:~# ip route
default via 1.1.1.1 dev eth0
1.1.1.0/24 dev eth0 proto kernel scope link src 1.1.1.3
root@test-ueb01:~# ping google.com
PING google.com (74.125.126.102) 56(84) bytes of data:
64 bytes from 74.125.126.102: icmp_seq=1 ttl=30 time=128 ms
64 bytes from 74.125.126.102: icmp_seq=2 ttl=30 time=119 ms
64 bytes from 74.125.126.102: icmp_seq=3 ttl=30 time=120 ms
64 bytes from 74.125.126.102: icmp_seq=4 ttl=30 time=119 ms
64 bytes from 74.125.126.102: icmp_seq=5 ttl=30 time=119 ms
--- google.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4000ms
rtt min/avg/max/mdev = 119.512/121.511/120.143/3.341 ms
root@test-ueb01:~#

```

Obrázek 4.8: Test konektivity PFSense

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



Obrázek 4.9: Ukázka NAT session

```
root@mnmg01:~# 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@mnmg01:~#
```

Obrázek 4.10: Fortigate VM intergace konfigurace

```

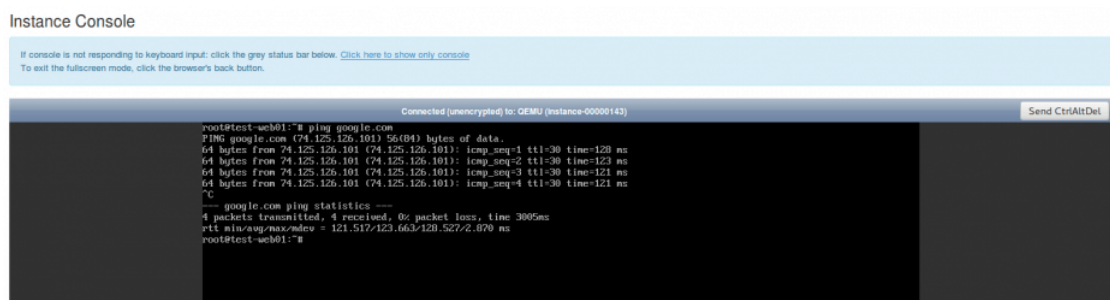
ubuntu@Management:~$ ssh root@172.0.0.5
Welcome to Ubuntu 14.04.3 LTS (GNU/Linux 3.19.0-26-generic x86_64)

 * 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] Network Functions Virtualisation, Dostupné online: [https://portal.etsi.org/NFV/NFV\\_White\\_Paper.pdf](https://portal.etsi.org/NFV/NFV_White_Paper.pdf)
- [2] 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>
- [3] 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>
- [4] 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>
- [5] 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>

# **Přílohy**



# Seznam obrázků

2.1	Základní znázornění virtualizace . . . . .	3
2.2	Vize Virtualizace síťových funkcí . . . . .	4
2.3	NFV architektura . . . . .	5
3.1	Popis heat orchestrace . . . . .	6
3.2	Testovací topologie . . . . .	7
4.1	Architektura NFV řešení . . . . .	9
4.2	Vytvořený pool . . . . .	10
4.3	Vytvoření members . . . . .	10
4.4	Vytvořený health monitor . . . . .	11
4.5	Vytvořená síťová topologie . . . . .	11
4.6	Test konektivity a load balancingu . . . . .	12
4.7	Síťová topologie . . . . .	13
4.8	Test konektivity PFSense . . . . .	13
4.9	Ukázka NAT session . . . . .	14
4.10	Fortigate VM intergace konfigurace . . . . .	14
4.11	Fortigate VM NAT konfigurace . . . . .	15
4.12	Test konektivity . . . . .	15

## 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	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: .....