

UNIVERZITA PARDUBICE

FAKULTA ELEKTROTECHNIKY A
INFORMATIKY

DIPLOMOVÁ PRÁCE

2024

BC. LUKÁŠ JANÁČEK

Univerzita Pardubice
Fakulta Elektrotechniky a Informatiky

Privátní cloud pro firmu střední velikosti
Diplomová práce

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2023/2024

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Bc. Lukáš Janáček**
Osobní číslo: **I22160**
Studijní program: **N0613A140007 Informační technologie**
Téma práce: **Privátní cloud pro firmu střední velikosti**
Zadávající katedra: **Katedra softwarových technologií**

Zásady pro vypracování

Cílem práce bude vytvoření a zprovoznění privátního cloudu na opensource platformě OpenStack. Práce bude obsahovat popis: typy cloudu, fungování OpenStacku, představení jednotlivých služeb (Keystone – autentizace, Horizon – webové rozhraní, Neutron – SDN v rámci OpenStack, NaaS pro komunikaci, Nova – škálovatelná služba pro přístup k infrastruktuře v rámci výpočtů, Glance – hledání, získávání a registrace virtuálních obrazů/strojů, Swift – objektové úložiště). Náplní praktické části bude zdokumentování kroků a nasazení OpenStacku a uvedených služeb.

Rozsah pracovní zprávy: **50**
Rozsah grafických prací:
Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam doporučené literatury:

AN OPENINFRA FOUNDATION PROJECT. The Most Widely Deployed Open Source Cloud Software in the World. Online. Dostupné z: <https://www.openstack.org/>. [cit. 2023-10-04].

MATTHEW HELMKE, Matthew. Ubuntu Linux: Unleashed 2021 Edition. 14. Addison-Wesley Professional, 2020. ISBN 978-0136778851.

Vedoucí diplomové práce: **Ing. Soňa Neradová, Ph.D.**
Katedra informačních technologií

Datum zadání diplomové práce: **8. listopadu 2023**

Termín odevzdání diplomové práce: **17. května 2024**

Ing. Zdeněk Němec, Ph.D. v.r.
děkan

L.S.

prof. Ing. Antonín Kavička, Ph.D. v.r.
vedoucí katedry

V Pardubicích dne 30. listopadu 2023

Prohlašuji:

Práci s názvem Privátní cloud pro firmu střední velikosti jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 7/2019 Pravidla pro odevzdávání, zveřejňování a formální úpravu závěrečných prací, ve znění pozdějších dodatků, bude práce zveřejněna prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 17. 5. 2024

Bc. Lukáš Janáček

ANOTACE

Diplomová práce se zabývá tématem zprovoznění a konfigurace softwarové platformy OpenStack pomocí instalačních skriptů DevStack, na serveru, který bude sloužit jako privátní cloud. Tento server pracuje v režimu ovládacího uzlu v rámci platformy OpenStack a je autoritativním serverem pro ostatní výpočetní uzly, pokud by se k tomuto serveru připojily. Konkrétněji se tato práce zabývá instalací platformy na operační systém Ubuntu Server, ve verzi 22.04 LTS a následnou konfigurací a otestováním vybraných modulů. Dále jsou popsány jednotlivé moduly softwarové platformy OpenStack včetně nástrojů na správu platformy OpenStack.

KLÍČOVÁ SLOVA

OpenStack, cloud, privátní cloud, server, Keystone, Horizon, Neutron, Nova, Glance, Swift.

TITLE

Private cloud for mid-sized business.

ANNOTATION

This master's thesis deals with the topic of commissioning and configuring of OpenStack software platform, with usage of DevStack install scripts. This platform is deployed onto a server that provides private cloud capabilities. This server is set up as controller node and is therefore authoritative for other compute nodes, that would be connected to this control server. This thesis focuses on the deployment onto an Ubuntu Server 22.04 LTS operating system and its subsequent configuration and testing of specific modules. Additionally, the modules that are a part of OpenStack platform are described in this thesis, including the tools for management of the OpenStack platform.

KEYWORDS

OpenStack, cloud, private cloud, server, Keystone, Horizon, Neutron, Nova, Glance, Swift.

PODĚKOVÁNÍ

Rád bych, na prvním místě, poděkoval Ing. Soně Neradové, Ph.D. za odborné vedení diplomové práce a za poskytnuté rady ohledně psaní této diplomové práce. Dále bych chtěl poděkovat svojí rodině za veškerou vyjádřenou podporu napříč navazujícím studiem.

OBSAH

Seznam obrázků	10
Seznam tabulek	11
Seznam příkazů	12
Seznam kódů	13
Seznam zkratk	14
Úvod	15
1 Software OpenStack	17
1.1 Použití.....	17
1.1.1 Požadavky na hardware.....	18
1.1.2 Požadavky na software.....	19
1.2 DevStack.....	19
1.3 Typy cloudů dle přístupu.....	20
1.3.1 Veřejný cloud.....	20
1.3.2 Privátní cloud.....	20
1.3.3 Hybridní cloud.....	20
1.4 Typy cloudů dle poskytovaných služeb.....	21
1.4.1 Infrastructure as a Service.....	21
1.4.2 Platform as a Service.....	21
1.4.3 Software as a Service.....	21
2 Moduly softwaru OpenStack	22
2.1 Moduly.....	22
2.1.1 Nova.....	22
2.1.2 Zun.....	23
2.1.3 Ironic.....	23
2.1.4 Cyborg.....	23
2.1.5 Swift.....	24
2.1.6 Cinder.....	24
2.1.7 Manila.....	24
2.1.8 Neutron.....	25
2.1.9 Octavia.....	25
2.1.10 Designate.....	26
2.1.11 Keystone.....	26
2.1.12 Placement.....	27
2.1.13 Glance.....	27
2.1.14 Barbican.....	28
2.1.15 Heat.....	28
2.1.16 Senlin.....	29
2.1.17 Mistral.....	29
2.1.18 Zaqar.....	29
2.1.19 Blazar.....	30
2.1.20 Aodh.....	30
2.1.21 Magnum.....	31
2.1.22 Sahara.....	31
2.1.23 Trove.....	32

2.1.24	Masakari.....	32
2.1.25	Murano.....	32
2.1.26	Solum.....	33
2.1.27	Freezer.....	33
2.1.28	EC2API.....	34
2.1.29	Horizon.....	34
2.1.30	Skyline.....	34
2.2	Operační moduly.....	35
2.2.1	Ceilometer.....	35
2.2.2	Monasca.....	35
2.2.3	Watcher.....	36
2.2.4	Vitrage.....	36
2.2.5	Adjutant.....	36
2.2.6	Cloudkitty.....	37
2.2.7	Rally.....	37
2.2.8	Tempest.....	37
2.3	Doplňky ke modulům.....	39
2.3.1	Storlets.....	39
2.4	Integrační nástroje.....	40
2.4.1	Kuryr.....	40
2.4.2	Tacker.....	40
3	Klientské nástroje.....	41
3.1	CLI.....	41
3.1.1	OpenStackClient.....	41
3.1.2	SDK.....	41
3.1.3	OpenStackSDK.....	41
4	Nástroje pro nasazení.....	42
4.1.1	OpenStack-Helm.....	42
4.1.2	Kolla-Ansible.....	42
4.1.3	Kayobe.....	42
4.1.4	OpenStack-Ansible.....	42
4.1.5	OpenStack-Charms.....	43
4.1.6	Bifrost.....	43
4.1.7	OpenStack-Chef.....	43
4.2	Sestavovací návody pro populární nástroje.....	43
4.2.1	LOCI.....	43
4.2.2	Puppet-OpenStack.....	43
5	Prvotní nastavení prostředí.....	45
5.1	Instalace operačního systému.....	45
5.1.1	Stažení obrazu ISO.....	45
5.1.2	Instalace operačního systému.....	46
5.2	Instalace prerekvizit a příprava uživatele.....	47
5.2.1	Nástroj git.....	47
5.2.2	Vytvoření uživatele stack.....	48
6	Instalace softwaru OpenStack.....	50
6.1	Stažení instalačního balíčku.....	50

6.1.1	Naklonování repozitáře softwaru OpenStack	50
6.1.2	Příprava instalace	51
6.2	Instalace softwaru	52
6.2.1	Spuštění instalace.....	52
6.2.2	Průběh a konec instalace.....	52
6.2.3	Sledování prostředků během instalace a běhu platformy	55
6.2.4	Dodatečná konfigurace úložiště.....	55
7	Keystone.....	57
7.1	Instalace	57
7.2	Spuštění.....	57
7.2.1	Otestování běhu	57
7.2.2	Parametry konfigurace	58
7.3	Správa projektů, uživatelů a rolí	60
7.3.1	Správa projektů	60
7.3.2	Správa uživatelů.....	62
7.3.3	Správa rolí.....	63
8	Horizon	65
8.1	Instalace	65
8.2	Spuštění.....	65
8.2.1	Potřebná konfigurace	66
8.2.2	Parametry konfigurace	66
8.3	Sledování stavu	66
8.3.1	Přihlášení	67
8.3.2	Přístup přes API.....	68
8.4	Propojení s ostatními komponenty	69
9	Nova.....	70
9.1	Instalace	70
9.2	Spuštění.....	70
9.2.1	Potřebná konfigurace	71
9.3	Nástěnka modulu Horizon	73
9.3.1	Administrativní nástěnka	73
9.3.2	Hypervizory	73
9.3.3	Host Aggregates.....	74
9.3.4	Instances.....	75
9.3.5	Flavors	76
9.4	Problémy s modulem Nova.....	76
9.4.1	Nemožnost spuštění jakékoliv instance	77
9.4.2	Nemožnost startu instancí po restartu systému.....	77
9.4.3	Nemožnost pokračování instance z jejího pozastavení.....	78
10	Glance	79
10.1	Instalace	79
10.2	Spuštění.....	79
10.2.1	Potřebná konfigurace	80
10.3	Práce s obrazy virtuálních strojů.....	81

10.3.1	Přidání.....	81
10.3.2	Správa metadat.....	82
10.3.3	Odebrání obrazů.....	83
11	Swift.....	84
11.1	Instalace	84
11.2	Spuštění.....	85
11.2.1	Parametry runtime konfigurace	85
11.3	Datové prostory.....	85
11.3.1	Vytvoření	85
11.3.2	Správa	86
11.3.3	Nahrávání souborů	87
12	Neutron	88
12.1	Instalace	88
12.2	Spuštění.....	88
12.2.1	Potřebná konfigurace	89
12.2.2	Start služby neutron	89
12.3	Sítě	90
12.3.1	Nástěnka modulu Neutron	90
12.3.2	Topologie sítě	90
12.3.3	Vytvoření sítě.....	91
12.3.4	Přiřazení sítě k instanci	92
	Závěr	94
	Použitá literatura	95
	SEZNAM PŘÍLOH.....	99

SEZNAM OBRÁZKŮ

Obrázek 1: Příklad běžícího programu htop v terminálu, po instalaci platformy OpenStack ..55	
Obrázek 2: Přihlašovací stránka nástěnky modulu Horizon.....67	67
Obrázek 3: Přehled platformy OpenStack v nástěnce modulu Horizon67	67
Obrázek 4: Autentizační údaje, vypsané v nástěnce modulu Horizon68	68
Obrázek 5: Naslouchající služby protokolu HTTP.....68	68
Obrázek 6: Nástěnka hypervizorů modulu Nova.....74	74
Obrázek 7: Metadata agregací jednotlivých uzlů a jejich filtrování.....75	75
Obrázek 8: Vytvořené instance modulu Nova.....75	75
Obrázek 9: Vytváření instancí virtuálních serverů v nástěnce modulu Nova.....76	76
Obrázek 10: Selhání služby n-cpu modulu Nova po startu systému77	77
Obrázek 11: Správa obrazů pro virtuální servery, pocházející z modulu Cinder81	81
Obrázek 12: Vytvoření a nahrání nového obrazu do modulu Glance.....82	82
Obrázek 13: Správa metadat v nástěnce modulu Glance.....83	83
Obrázek 14: Probíhající klonování repozitáře modulu Swift84	84
Obrázek 15: Vytvoření nového kontejneru modulu Swift ve vybraném projektu.....86	86
Obrázek 16: Správa kontejnerů ve specifickém projektu86	86
Obrázek 17: Nahrání nového souboru společně s jeho hierarchickou cestou.....87	87
Obrázek 18: Nástěnky modulu Neutron v nástěnce Horizon90	90
Obrázek 19: Přehled topologie sítí v nástěnce modulu Neutron91	91
Obrázek 20: Vytvoření sítě modulu Neutron.....92	92
Obrázek 21: Připojená síť modulu Neutron k instanci virtuálního serveru92	92
Obrázek 22: Kontextová nabídka pro připojení nového rozhraní.....93	93
Obrázek 23: Přidání nového rozhraní a jeho připojení do sítě modulu Neutron93	93

SEZNAM TABULEK

Tabulka 1: Minimální a doporučená hardwarová výbava	18
Tabulka 2: Počet databázových dotazů provedených jednotlivými moduly během instalace..	54

SEZNAM PŘÍKAZŮ

Příkaz 1: Nahrání instalačního obrazu na USB flash disk v Linuxovém systému	46
Příkaz 2: Instalace nástroje git pomocí balíkových nástrojů.....	48
Příkaz 3: Vytvoření samostatného uživatele stack	48
Příkaz 4: Přepsání práv domovského adresáře uživatele stack.....	48
Příkaz 5: Přidání uživatele stack do souboru sudoers s editorem nano	49
Příkaz 6: Přepnutí na uživatele stack	49
Příkaz 7: Stažení instalačního repozitáře DevStack	50
Příkaz 8: Kontrola existence složky staženého repozitáře.....	50
Příkaz 9: Přepnutí do složky devstack a vytvoření konfigurace hesel.....	51
Příkaz 10: Spuštění instalace platformy OpenStack	52
Příkaz 11: Vytvoření nového disku LVM s oddílem cinder-volumes	56
Příkaz 12: Příkaz požadavku HTTP na spuštěný modul Keystone	58
Příkaz 13: Načtení nových hodnot prostředí pomocí příkazu source	59
Příkaz 14: Načtení nových proměnných pomocí opětovného přihlášení	59
Příkaz 15: Test příkazu openstack s nastavenými proměnnými prostředí.....	59
Příkaz 16: Test příkazu openstack s nastavenými proměnnými prostředí.....	60
Příkaz 17: Založení nového projektu v modulu Keystone.....	61
Příkaz 18: Vypsání vlastností projektu s použitím projektového id	61
Příkaz 19: Změna vlastností projektu s použitím projektového id a nového jména	61
Příkaz 20: Vymazání projektu s použitím projektového id	61
Příkaz 21: Příkaz openstack pro výpis vytvořených uživatelů	62
Příkaz 22: Vytvoření nového uživatele.....	62
Příkaz 23: Zapnutí a vypnutí uživatele	63
Příkaz 24: Modifikace údajů uživatele	63
Příkaz 25: Odstranění uživatele	63
Příkaz 26: Výpis rolí.....	63
Příkaz 27: Vytvoření nové role	64
Příkaz 28: Přiřazení a odebrání role od uživatele	64
Příkaz 29: Vytvoření a odebrání implikovaných rolí.....	64
Příkaz 30: Vynucený restart instance, která se není schopna spustit příkazem Resume	78
Příkaz 31: Vytvoření a přidání uživatele glance a jeho rolí, pro instalaci modulu Glance	79
Příkaz 32: Přeinstalace softwaru OpenStack	84
Příkaz 33: Vytvoření a přidání uživatele neutron a sítí, pro instalaci modulu Neutron	88
Příkaz 34: Finální příprava databáze modulu Neutron	89

SEZNAM KÓDŮ

Kód 1: Nový záznam uživatele stack v souboru /etc/sudoers.....	49
Kód 2: Uložená hesla v souboru local.conf.....	51
Kód 3: Výstup sledovaných časování během instalace vč. grafu.....	53
Kód 4: Finální výstup instalačních informací po dokončení instalace.....	54
Kód 5: Vyžadované proměnné prostředí pro práci s příkazem openstack.....	58
Kód 6: Automatizace proměnných pomocí souboru .bash_profile.....	58
Kód 7: Konfigurační proměnné nástěnky Horizon.....	66
Kód 8: Konfigurace vlastních parametrů ovládacího uzlu modulu Nova.....	71
Kód 9: Konfigurace parametrů Keystone u ovládacího uzlu modulu Nova.....	72
Kód 10: Konfigurace parametrů Glance a Placement u ovládacího uzlu modulu Nova.....	72
Kód 11: Konfigurace parametrů Glance a Placement u ovládacího uzlu modulu Nova.....	77
Kód 12: Vkládané parametry pro konfiguraci modulu Glance.....	80
Kód 13: Konfigurace parametrů Keystone u ovládacího uzlu modulu Glance.....	80
Kód 14: Přidání požadavku na instalaci modulu Swift v souboru local.conf.....	84
Kód 15: Ukázka parametrů konfigurace objektového úložiště modulu Swift.....	85
Kód 16: Vkládané parametry pro konfiguraci metadat modulu Neutron.....	89
Kód 17: Vkládané parametry modulu Neutron pro konfigurační soubor modulu Nova.....	89

SEZNAM ZKRATEK

API	Application Programming Interface
ASIC	Application Specific Integrated Circuit
AWS	Amazon Web Services
CD	Compact Disk
CLI	Command Line Interface
CPU	Central Processing Unit
DPU	Data Processing Unit
DVD	Digital Video Disk
FPGA	Field Programmable Gate Array
GPU	Graphics Processing Unit
HDD	Hard Disk Drive
IaaS	Infrastructure as a Service
IPMI	Intelligent Platform Management Interface
ISO	ISO 9660 Image
MQ	Message Queue
NFV	Network Function Virtualization
NIC	Network Interface Card
OCI	Open Container Initiative
PaaS	Platform as a Service
PXE	Preboot Execution Environment
RAID	Redundant Array of Inexpensive Disks Lze také použít Redundant Array of Independent Disks
RAM	Random Access Memory
REST	Representational State Transfer
SaaS	Software as a Service
SDK	Software Development Kit
SDN	Software Defined Networking
SSD	Solid State Drive
SSH	Secure Shell
SQL	Structured Query Language
USB	Universal Serial Bus

ÚVOD

Cílem této práce je seznámit čtenáře se softwarem OpenStack (originálně stylizováno jako openstack), společně s jeho nasazením na výpočetní platformu. Rozsahově se jedná o nasazení softwaru pro typicky firemní potřeby, a to o střední velikosti rozsahu působnosti.

V teoretické části je popsán celý ekosystém platformy OpenStack včetně všech jeho modulů zveřejněných k datu psaní této práce. Každá implementace platformy OpenStack se skládá ze základních modulů (Keystone, Nova, Glance, Cinder a Neutron) a z mnoha dalších modulů, které rozšiřují funkcionalitu a umožňují tak uživatelům přizpůsobit si platformu podle svých specifických potřeb. Pro každý modul je uvedena jeho funkce, oblast zaměření a seznam závislých modulů. Kromě toho je pro některé moduly uveden krátký přehled s čím konkrétně modul pracuje, na jakých principech operuje a případně čím se liší moduly, které pracují v podobných oblastech zaměření.

V praktické části jsou podrobně, podrobně popsány a přiblíženy postupy k instalaci celého softwaru OpenStack, konkrétně v režimu DevStack pro osobní, testovací nebo vývojové účely. Vyšší důraz je věnován službám Keystone, Horizon, Neutron, Nova, Glance a Swift, které jsou blíže rozebrány i v teoretické části. Instalační postup je vedle detailních popisků také doprovázen ukázkami kódu, jenž byly použity pro instalaci softwaru OpenStack a konfiguraci softwaru OpenStack společně s operačním systémem tak, aby bylo možné plnohodnotně uvést do provozu celou softwarovou platformu.

Teoretická část

1 SOFTWARE OPENSTACK

Software OpenStack je cloudová platforma, jinak také mylně označována za „cloudový operační systém“, sloužící pro ovládání, dohlížení a rozprostření jednotlivých prostředků v datových centrech. Cílem je dát administrátorům systému ucelený kontrolní bod, v podobě webového rozhraní, pro dohled, alokaci a řízení prostředků a zajistit tak jejich dostupnost v rámci architektury IaaS.

[1]

1.1 Použití

Software OpenStack lze použít a škálovat pro virtualizaci jakéhokoliv množství prostředků v rámci výpočetní platformy. Speciální důraz celého projektu je kladen na sjednocenou kontrolu mnoha výpočetních stanic, například v datových centrech, kde použití sjednoceného ovládacího prvku pro řízení vysokého množství je vysoce žádoucí a přináší zjednodušení práce s celým systémem.

Jednotlivé moduly využívají systém různými způsoby, kde některé moduly ulehčují přístup ke sdíleným datům na paměťových úložištích, jiné zase zpracovávají „orchestraci“ virtuálních systémů. Rozšířená působnost platformy OpenStack umožňuje použití takového systému pro mnoho různých systémových požadavků, které se liší dle požadavků v rámci poskytovaných služeb.

[1], [2]

1.1.1 Požadavky na hardware

Dle oficiální dokumentace nejsou specifikovány jakékoliv konkrétní požadavky na hardware kromě požadované architektury, a to konkrétně x86 (ideálně avšak x86_64, jinde také označované jako amd64). Požadavky se liší dle jednotlivých uzlů a modulů, které na nich pobeží, společně s minimálními požadavky na operační systém. Nicméně lze z požadavků jednotlivých modulů sestavit tabulku minimálních a doporučených požadavků na hardware, s ohledem na moderní systémy se současnou technologií:

Tabulka 1: Minimální a doporučená hardwarová výbava

Komponent	Minimální hardware	Doporučený hardware
CPU	4 jádrový s frekvencí 2,4 GHz	24 jádrový s frekvencí 2,67 GHz
RAM	8 GB	24 GB nebo více
HDD	2x 500 GB (7200 ot/min)	4x 500 GB (7200 ot/min)
SSD	1x 500 GB	2x 500 GB
RAID	Softwarový RAID-1 (se specifickým použitím softwaru mdadm kvůli zvýšení rychlosti čtení v režimu RAID-1)	Hardwarový RAID-10

Nutno je také podotknout, že požadavky na hardware se budou lišit dle specializací jednotlivých uzlů. Například pro uzly obstarávající úložný prostor bude žádoucí vyšší počet rychlých disků a hardwarových ovladačů pro řízení RAID polí, zatímco výpočetní uzly budou požadovat vyšší počet CPU nebo jejich jader společně s dalšími přídatnými kartami pro akceleraci výpočtů na GPU nebo jiných specializovaných kartách.

Jakékoliv dodatečné speciální požadavky na hardware specializovaných uzlů jsou uvedeny v textu podkapitol jednotlivých modulů.

[1], [2]

1.1.2 Požadavky na software

Požadavky na software jsou v oficiální dokumentaci více přiblíženy. Oficiální požadavky zahrnují nástroje, programy a knihovny, které jsou běžně dostupné ke stažení v balíkovacích systémech v rámci operačních systémů, na kterých běží jednotlivé uzly. Oficiálně podporované a otestované operační systémy jsou následující:

- SUSE Linux Enterprise Server,
- openSUSE,
- Red Hat Enterprise Linux (často stylizován/zkrácen jako RHEL),
- CentOS,
- Ubuntu.

Jednotlivé verze softwaru OpenStack mají různou podporu pro různé verze těchto operačních systémů. Společně s tímto je nutné uvést upozornění, že novější verze operačních systémů typicky nemají určité verze softwaru OpenStack. Například nejnovější Ubuntu 23.10 nemusí obsahovat zpětnou podporu pro starší nebo i nejnovější verze softwaru OpenStack. Různé repozitáře umožňují přístup k tzv. archivním verzím, takové verze většinou nebývají aktualizovány.

Veškeré požadované balíčky s prerekvizitami pro instalaci softwaru OpenStack, a to i ve verzi DevStack lze typicky automatizovaně nainstalovat za pomoci instalačních skriptů, blíže popsanych v praktické části.

[1], [2], [3], [4]

1.2 DevStack

Software DevStack je sada skriptů a nástrojů pro automatizované nasazení softwaru OpenStack. Cílem je docílit možností rychlého nasazení softwaru OpenStack do čistého/nově nainstalovaného operačního systému společně s celou řadou různých cílů pro usnadnění vývoje, testování, ověření konfigurací či testování kooperace jednotlivých modulů. V rámci praktické části bude použita právě verze DevStack pro instalaci a konfiguraci platformy OpenStack na nově nainstalovaný operační systém.

[2], [3], [4]

1.3 Typy cloudů dle přístupu

V praxi lze rozlišit určité typy cloudů, lišící se typicky jejich zaměřením, infrastrukturou, nad kterou jsou nasazeny, a jejich rozsahem.

1.3.1 Veřejný cloud

Cloud, který je dostupný veřejnosti a třetím stranám, typicky poskytován buď v základní formě jako služba zdarma, či podle možnosti finančních prostředků, škálovatelný dle požadavků zájemce. Často se také vyskytují určité úrovně, různě zpoplatněné, které se liší dle zaměření či kapacity prostředků. Jeden takový typ cloudů může poskytovat úložiště pro zákazníky a slouží pouze k tomuto účelu, zatímco další typ může poskytovat služby VPS, kde má uživatel přístup k vlastnímu běžícími virtuálnímu serveru, přímo zapojeném v internetové síti, nejčastěji s vlastní veřejnou adresou IP.

[5], [6]

1.3.2 Privátní cloud

Typ cloudu, který je vymezen pro potřeby jedné společnosti nebo skupiny uživatelů, kteří mají exkluzivní přístup k takovému cloudu. Tyto cloudy slouží nejčastěji pro potřeby vlastního běhu služeb, které jsou následně prodávány danou společností jako služba. Přístup je silně omezen na určité segmenty. Tento cloud se typicky nachází na vlastním místě, kde je společností provozován v prostorách, spadající pod danou společnost, aby se zamezilo přístupu ostatních třetích stran a aby nedocházelo k úniku dat provozovatelem cloudu, který by běžel na serverech jiné společnosti.

[5], [6]

1.3.3 Hybridní cloud

Tato forma cloudu se vyskytuje v internetu nejčastěji. Jedná se o spojení obou typů cloudů, veřejného a privátního, který je nejčastěji rozdělen na dva segmenty, kde řídicí část cloudu je spravována jednou společností, která prodává části tohoto cloudu jako službu třetím stranám.

[5], [6]

1.4 Typy cloudů dle poskytovaných služeb

Cloudy lze také dělit podle toho, co dané cloudové platformy nabízejí. Může se jednat o různé úrovně kontroly či zaměření. Zákazníkům to může ulehčit práci s konfigurací cloudu a mohou využívat například pouze úložnou část cloudové infrastruktury, bez dodatečného ladění.

1.4.1 Infrastructure as a Service

Tento typ cloudu, zkrácen nejčastěji na IaaS, se zaměřuje na poskytování kompletní infrastruktury jako službu, která je poskytována. Příkladem může být pronájem serverů VPS, kde mají uživatelé plný přístup ke kompletnímu systému, tak, jak by běžel na běžném osobním počítači či serveru. Využití takového cloudu lze najít v rámci provozu virtuálních strojů, úložiště, provozu sítí, kde je zapotřebí vzdálený či oddělený přístup nebo běh různých systémů, a to i na různých hardwarových architekturách.

[5], [6]

1.4.2 Platform as a Service

Cloudy tohoto typu jsou poskytovány jako zhotovené prostředí pro specifický účel. Nejčastěji se lze setkat s využitím takovýchto cloudů jako s výpočetními stanicemi pro specifické úkony, například sestavování či kompilace softwaru, jeho testování a přístup k testovací databázi, běžící na stejném serveru. Stejně jako cloudy typu IaaS mohou tyto servery, běžící v cloudu, sloužit k vývoji, otestování, sestavení a zabalení aplikací či programů na různých hardwarových architekturách, kde vývoj ve vlastním prostředí je buďto nemožný či velmi nákladný.

[5], [6]

1.4.3 Software as a Service

Tento typ cloudu, ačkoliv značně omezený ve své působnosti, je silně specializován na poskytování softwaru za pomoci internetového připojení, kde je žádoucí udržovat dané aplikace či programy aktualizované a zároveň sjednocené ve způsobu jejich používání.

[5], [6]

2 MODULY SOFTWARE OPENSTACK

Software OpenStack se skládá z mnoha modulů, jenž se starají o různé funkce v rámci platformy. V současné verzi softwaru OpenStack je možné využít 41 modulů, poskytující různé služby.

[1], [2]

2.1 Moduly

Každý modul je naprogramován s ohledem na jednu činnost nebo velmi specializované pole působnosti. Platforma OpenStack umožňuje spouštět pouze explicitně požadované služby, aby se předešlo neefektivní práci se systémovými zdroji.

Některé moduly také zpřístupňují tzv. API, přístupový bod pro získávání dat z modulu. Typické využití se najde při automatizovaném získávání dat, ať už z ostatních modulů, webových aplikací nebo specializovaných programů, tak i přímé napojení na služby třetích stran jako například Amazon nebo Microsoft Azure.

[1], [2]

2.1.1 Nova

Jeden ze základních modulů, starající se o distribuci prostředků napříč infrastrukturou. Stará se o spouštění, běh a ukončování virtuálních serverů, společně s jejich administrací. Celý modul je založen na tzv. „on-demand“ architektuře, kde veškeré požadavky jsou zpracovávány až po jejich přijetí. Nova se tedy stará o efektivní využití zdrojů napříč celou infrastrukturou.

Ve výchozím nastavení využívá modul Nova virtualizační podporu jádra operačního systému. Nova běží na výpočetních uzlech, kde samotný modul Nova přistupuje k architektuře hypervizora, běžícího pod operačním systémem, pomocí kterého řídí jednotlivé virtuální servery. Výpočetní uzel, na který je modul Nova nasazován musí mít operační systém i hardware s podporou virtualizace.

Požadavkem pro spuštění modulu Nova na výpočetním uzlu je přítomnost alespoň 2 síťových karet. Takovýchto uzlů se může v infrastruktuře nacházet mnoho, ale vždy musí být spojeny s platformou v jedné společné infrastruktuře. Tyto síťové karty jsou poté využity pro připojení virtuálních sítí a k jejich obsluze.

Modul Nova je závislý na modulech Keystone, Neutron, Glance a Placement.

[1], [2], [7]

2.1.2 Zun

Kontejnerový modul Zun poskytuje API pro spouštění a správu kontejnerů. Cílem je usnadnění práce s kontejnery a zavést tak abstrakční úroveň, u které není nutné do podrobnosti znát jaké kontejnerové a virtualizační technologie běží na pozadí modulu Zun.

Celkově se jedná o modul, který blíže integruje kontejnery do platformy OpenStack. Například přímým zapojením do sítě modulu Neutron nebo oddílů modulu Cinder. Toho dociluje za pomoci zpřístupnění API, pomocí které lze spravovat veškeré aspekty modulu Zun.

Modul Zun je závislý na modulech Keystone, Neutron a Kuryr.

[1], [2], [7]

2.1.3 Ironic

Modul Ironic má za cíl přidělování a řízení přímého přístupu k hardwaru, virtuálním serverům a kontejnerům. Tento modul lze použít i nezávisle na platformě OpenStack, tak i v rámci integrace ve sdílené infrastruktuře. Cílem je tedy umožnit práci s přímým hardwarem rozdílných serverů, jako by se jednalo servery virtuální.

Správa hardwaru je nejčastěji uskutečněna pomocí již existujících technologií vzdálené správy. Nejčastěji se lze setkat s běžnými technologiemi IPMI a PXE nebo také s proprietárním řešením různých výrobců, například společností HP (iLO), Dell (iDRAC), Cisco (IMC) nebo Intel (AMT). Typicky jsou tyto proprietární nástroje zčásti založeny na technologii IPMI.

Pokud se rozhodneme využít modul Ironic v rámci sdílené infrastruktury platformy OpenStack, pak se pracuje s těsnou integrací s moduly Keystone, Nova, Neutron, Glance a Swift.

[1], [2], [7], [9], [10], [11]

2.1.4 Cyborg

Modul Cyborg poskytuje jednolitý framework pro použití a řízení specializovaných hardwarových akceleratorů, které mají sloužit k velmi vysokému zrychlení výpočtů ve vymezené škále použitelnosti.

Takové akcelerátory se nacházejí v mnoha sjednocených variantách, kde pravděpodobně nejznámějším zástupcem je běžná, komerčně dostupná grafická karta (GPU). Dále se může jednat o specializované hardwarové moduly založené na polích FPGA, na míru vyráběné hardwarové urychlovače typu ASIC nebo dnešní moderní úzce specializovaný hardware

například pro inferenci modelů umělých inteligencí nebo hromadné zpracování operací v datacentrech, kolektivně sdružený pod zkratkou DPU (Data Processing Unit).

Samotný modul Cyborg je nadále závislý na modulech Keystone a Glance.

[1], [2], [7], [12]

2.1.5 Swift

Modul pro správu objektového úložiště. Cílem modulu Swift je zajistit plně distribuované a konzistentní úložiště s vysokou dostupností a odolností. Toto úložiště se lehce specializuje na ukládání nesourodých dat, u kterých není předem znám jejich objem nebo počet. Celé úložiště je navrženo pro efektivní škálování a optimalizovaný přístup, dostupnost a redundanci.

Vnitřně tento modul používá tzv. těžítka pro přiřazování úložných prostorů pro specifická data. Společně s určováním umístění jednotlivých položek nebo skupin dat také zajišťuje jejich replikaci, a to v rámci celé infrastruktury. Typicky pro zrychlení práce takového úložiště je používána replikace jak k účelům zvýšení odolnosti, tak i k značnému urychlení přístupu k datům, kde se jednotlivé repliky dat ukládají co nejbliže k místu jejich ukládání a použití.

Tento modul není závislý na jiných modulech ani není úzce spojen z žádným z nich.

[1], [2], [7]

2.1.6 Cinder

Modul, se specializací na blokové úložiště, které zpřístupňuje v rámci platformy OpenStack. Abstrahuje správu blokových úložišť, jako jsou pevné disky. Podílí se také na distribuci paměťových oddílů pro virtuální servery modulu Nova, přímým hardwarovým uzlům modulu Ironic nebo kontejnerům modulu Zun.

Cíle modulu Cinder jsou zprostředkování komponentní architektury pro snadné přidání nových oddílů a úložišť, vysoká dostupnost společně s vysokou odolností, dodatečnou obnovitelnost při výpadku částí úložiště a zprostředkováním otevřené API pro práci s úložišti a modulem Cinder.

Tento modul je závislý na modulu Keystone.

[1], [2], [7]

2.1.7 Manila

Třetí modul, přímo pracující s úložištěm, zprostředkovává koordinovaný a sjednocený přístup k heterogenním souborovým systémům. Cílem je umožnit přístup k co nejvyššímu počtu

souborových systémů a umožnit tak použití vyhovujícího souborového systému pro daný případ použití. Společně s faktem určitých nekompatibilit mezi souborovými systémy a operačními systémy, které s nimi pracují, je vysoce žádoucí u takovéto infrastruktury zajistit sjednocený přístup do paměťových úložišť. Například operační systém Microsoft Windows neumožňuje čtení oddílů typu ext4, který je primárně využíván operačními systémy založenými na Linuxu, zatímco linuxové operační systémy umí číst i zapisovat do oddílů typu NTFS, používaných na operačním systému Microsoft Windows.

Stejně jako předchozí moduly Swift a Cinder, pro práci s úložišti se snaží zajistit komponentovou architekturu, vysokou dostupnost i odolnost, obnovitelnost a otevřenou API.

Modul Manila není závislý na jiných modulech.

[1], [2], [7]

2.1.8 Neutron

Neutron je modul starající se o sestavení a provoz sdílené sítě napříč infrastrukturou. Celý projekt modulu Neutron je založen na principu SDN, softwarově definované sítě. Ta umožňuje centralizovaně řídit, monitorovat a měnit virtuální topologii sítě z řídicího uzlu, typicky označovaného jako centralizovaný ovládací prvek (v anglické literatuře controller).

Cílem je poskytnout síť jako službu, ve zkratce NaaS (Network-as-a-Service). Toho dosahuje za pomoci vytváření a správy virtuálních síťových rozhraní (vNIC), které kompletně abstrahují práci s reálnými, fyzickými síťovými rozhraní. Modul Neutron umožňuje pro řízení používat síťovou API, kterou lze zpřístupnit pro potřeby automatizace.

Modul Neutron je závislý na modulu Keystone.

[1], [2], [7]

2.1.9 Octavia

Modul Octavia se stará o rozložení zátěže napříč sítí v infrastruktuře. Původně se jednalo o samostatný, otevřený software, který byl designován, aby pracoval v rámci platformy OpenStack. Původně se vyvíjel tento modul s cílem poskytovat službu rozkládání zátěže jako službu, LBaaS (Load-Balancer-as-a-Service).

Rozložení zátěže napříč sítí je u škálovatelné infrastruktury velmi žádoucí až nutné, aby nedošlo k degradaci a snížení efektivity infrastruktury. Modul Octavia tohoto cíle dosahuje pomocí spouštění síťových kontrolérů na vyžádání. Když se síť začne blížit k maximální kapacitě tak

je modulem Octavia vytvořena nová instance amfory (server, na kterém běží Octavia kontroler), která je spuštěna buď v rámci virtuálního serveru, serveru na přímém hardwaru nebo kontejneru.

Modul Octavia je závislý na modulech Glance, Keystone, Neutron a Nova.

[1], [2], [7]

2.1.10 Designate

Modul Designate se v rámci platformy OpenStack stará o službu DNS, kterou poskytuje jak celé platformě, tak i infrastruktuře na které běží. Cílem je poskytovat DNS jako službu, DaaS (DNS-as-a-Service).

Cíle dosahuje pomocí rozložení zátěže a vytváření vícero DNS serverů v tzv. multi-tenantní architektuře. Veškeré DNS servery modulu Designate sdílí jednu databázi a úložiště, ale pro zrychlení celé infrastruktury poskytuje službu DNS na vícero místech, aby došlo k rozložení zátěže a výsledným snížením odezvy. V rámci poskytování DNS umožňuje Designate použít již existující DNS serverové programy, například Bind9 nebo PowerDNS 4. Pro práci s DNS systémem a autentizaci poskytuje modul Designate tzv. REST API pro přístup k autentizaci v modulu Keystone.

Modul Designate je závislý na modulu Keystone.

[1], [2], [7]

2.1.11 Keystone

Modul Keystone poskytuje službu pro identitu a autentizaci klientů. Jedná se jeden z centrálních modulů platformy OpenStack. Cílem modulu Keystone je centrálně řídit systém autentizace v rámci celé infrastruktury.

Pro veškeré ostatní moduly poskytuje modul Keystone API, přes kterou lze požádat o autentizaci a získat tím potvrzení, zdali se jedná o autentický požadavek. Zároveň se modul Keystone stará o správu těchto údajů, mezi které patří následující:

- uživatelé,
- skupiny,
- zdroje,
- projekty,

- domény,
- úkoly,
- role a jejich přiřazení
- tokeny,
- katalogy.

Podobně jako modul Designate pracuje modul Keystone v multi-tenantní architektuře, aby došlo k rozložení zátěže a snížení odezvy na požadavky.

Modul Keystone je jedním z pilířů platformy OpenStack a není závislý na žádném jiném modulu.

[1], [2], [7]

2.1.12 Placement

Jedná se o modul, starající se o rozdělování a sledování prostředků v cloudovém prostředí platformy OpenStack. Cílem je udržovat tyto informace a umožnit ostatním modulům efektivně pracovat s infrastrukturou a poskytovat dodatečné informace kde jsou a kam se mají alokovat prostředky.

Pro práci s modulem Placement je dostupná REST API, přes kterou je možné posílat požadavky na práci se zdroji. Zároveň modul Placement pracuje s definovatelnými zdroji, vnitřně označovanými jako třídy (classes). Tyto třídy obsahují přiřazovatelné hardwarové prostředky, typicky pojmenované podle zdroje, který poskytují, například DISK_GB, MEMORY_MB, VCPU nebo další, uživatelsky definovatelné třídy. Dodatečně k třídám poskytuje také tzv. vlastnosti, které blíže určují specifikovanou třídu, například disk může být s plotnový (HDD) nebo se může skládat z flash paměti (SSD).

Modul Placement je jedním z hlavních pilířů platformy OpenStack a není závislý na jiných modulech.

[1], [2], [7]

2.1.13 Glance

Glance je modul, který sleduje, zjišťuje, registruje a řídí práci s obrazy virtuálních strojů. Pro potřeby zprovoznění virtuálních serverů výpočetních uzlů nebo řízení infrastruktury, například moduly Nova, Cyborg nebo Octavia, poskytuje centralizovanou správu těchto obrazů.

Pro poskytování, předávání a řízení poskytuje modul Glance REST API, umožňující získávání metadat ohledně spravovaných obrazů. Pro čtení nebo ukládání těchto obrazů lze použít různá bloková, objektová nebo sdílená úložiště, které jsou poskytována jinými moduly v platformě OpenStack.

Modul Glance je závislý na modulu Keystone.

[1], [2], [7]

2.1.14 Barbican

Modul starající se o bezpečné uložení, ochranu a přístup ke sdíleným zabezpečeným datům napříč platformou OpenStack. Cílem je řídit přístup k velmi citlivým datům, typicky souvisejícím se zabezpečením infrastruktury.

Typická citlivá data nejčastěji zahrnují:

- symetrické šifrovací klíče,
- asymetrické šifrovací klíče,
- bezpečnostní certifikáty,
- binární data.

Pro poskytování těchto citlivých údajů používá vnitřně, pro zpracování přístupu, role a jejich přiřazení, na jejichž základě schvaluje přístup. V základním nastavení pracuje s rolemi:

- admin,
- creator,
- observer,
- audit,
- key-manager:service-admin.

Modul Barbican je závislý na modulu Keystone.

[1], [2], [7]

2.1.15 Heat

Modul Heat se stará o orchestrování zdrojů v rámci platformy OpenStack. Dosahuje toho pomocí šablon v podobě textových souborů, které jsou interpretovány jako kód. Dále lze tyto

šablony použít pro automatizované škálování poskytovaných služeb a sběr telemetrie, což umožňuje použít tyto škálovatelné šablony jako jednotlivé zdroje.

Tyto šablony specifikují vztahy mezi přiřaditelnými zdroji, například určitý diskový svazek náleží specifickému serveru. Informace o šablonách lze získat pomocí nativního přístupu pomocí OpenStack REST API nebo pomocí Query API pro platformu CloudFormation.

Modul Heat je závislý na modulu Keystone.

[1], [2], [7]

2.1.16 Senlin

Tento modul operuje nad tzv. clustery, skupinou homogenních zdrojů, které operují jako celek. Tyto skupiny zdrojů poté poskytuje ostatním modulům napříč infrastrukturou. Cílem je orchestrovat tyto skupiny stejných zdrojů co nejvíce efektivně.

Na rozdíl od modulu Heat, který orchestruje shluky jednotlivých zdrojů, orchestruje modul Senlin clustery, občas označované jako kolekce, identických kategorií, které poté poskytuje v rámci infrastruktury.

Modul Senlin je závislý na modulu Keystone.

[1], [2], [7]

2.1.17 Mistral

Pro správné zpracování postupných úloh, v anglické literatuře popsané pomocí slova workflow. Modul Mistral má za úkol zajistit správný postup provádění těchto postupů, za pomoci zdrojů dostupných v infrastruktuře.

Konfigurace je založena na souborech v jazyce YAML. A typicky slouží k automatizaci administrativních úloh, které jsou časově náročné a trvalo by dlouhou dobu, než by administrátor spustil další krok požadované úlohy.

Modul Mistral je závislý na modulu Keystone.

[1], [2], [7]

2.1.18 Zaqar

O předávání zpráv, typicky pro mobilní a webové rozhraní, se stará modul Zaqar. Poskytuje plnohodnotnou RESTful API, kterou lze použít pro přeposílání zpráv mezi různými komponenty platformy OpenStack.

Jedná se o multi-tenantní službu poskytující jednu z hlavních komponent pro fungování platformy a poskytování infrastruktury jako SaaS (Software as a Service). Použitím tohoto modulu lze docílit rozesílání zpráv přes různé komunikační kanály, například pro doručení do mobilních zařízení přes cloudovou službu.

Modul Zaqar je závislý na modulu Keystone.

[1], [2], [7]

2.1.19 Blazar

Blazar je modul, který se stará o rezervaci prostředků. Umožňuje jednotlivým instancím serverů nebo služeb v rámci platformy OpenStack rezervovat předem specifikované zdroje a jejich dostupného množství, a to buď na určitou dobu nebo v případě nutnosti i na neurčito.

Přínosem modulu Blazar je zlepšené monitorování jednotlivých zdrojů, a to současně využitých i do budoucna plánovaných, dynamické plánování na metrikách zátěže (včetně předpovídaného zatížení) a automatizaci přiřazování zdrojů a jejich následné vrácení.

V době vydání této práce (společně s verzí platformy OpenStack 2023.2) podporuje modul Blazar rezervaci následujících zdrojů:

- zdroje modulu Nova,
 - výpočetní uzly (hypervizory),
 - instance (servery)
- zdroje modulu Neutron,
 - plovoucí IP adresy.

Modul Blazar je závislý na modulech Keystone a Nova.

[1], [2], [7]

2.1.20 Aodh

Tento modul se stará o reagování na sledovaných akcí v modulu Ceilometer. Podle nadefinovaných pravidel a s nimi spojenými spouštěmi (trigger), modul Aodh reaguje na data, která jsou sesbírána modulem Ceilometer. Může se jednat o pouhé datové vzorky nebo události, která jsou vyvolána sesbíranými daty.

Modul Aodh je závislý na modulu Keystone.

[1], [2], [7]

2.1.21 Magnum

Modul Magnum se zabývá orchestrací kontejnerů v rámci infrastruktury. Pracuje, nad již vytvořenými technologiemi, které jsou schopné pracovat samostatně a abstrahuje práci s nimi, společně s běžícími kontejnery. Díky tomu poskytuje modul Magnum tyto kontejnery v rámci infrastruktury jako dostupné a přiřaditelné zdroje.

Pro práci s orchestračními nástroji používá modul Magnum dva typy objektů:

- Cluster,
- ClusterTemplate.

Kde objekty typu Cluster jsou zpracovávány jako kolekce uzlů, nad kterými jsou úlohy prováděny. Zatímco objekty typu ClusterTemplate jsou šablony pro takovéto kolekce, které umožňují mnohem rychlejší nasazení nových kolekcí na již stávající vytvořené infrastruktury kontejnerových nástrojů. V době vydání této práce podporuje modul Magnum následující orchestrační nástroje:

- Docker Swarm,
- Kubernetes,
- Apache Mesos.

Modul Magnum je závislý na modulech Keystone, Heat, Nova, Neutron a Glance.

[1], [2], [7]

2.1.22 Sahara

Sahara je modul spojující ostatní frameworky spolu s platformou OpenStack. Použit lze například následující frameworky:

- Apache Hadoop,
- Apache Spark,
- Apache Storm.

Konfigurace těchto frameworků probíhá pomocí přesného specifikování konfiguračních parametrů, zejména poté parametry určující topologii clusterů, typ a verze frameworků,

specifické uzly, na kterých běží cluster, společně s jejich detaily ohledně hardwaru a mnoho dalších.

Modul Sahara je závislý na modulech Cinder, Keystone, Glance, Heat, Neutron a Nova.

[1], [2], [7]

2.1.23 Trove

Trove je modul poskytující databáze jako službu (DaaS). Designován tak, aby běžel jako samostatný modul v platformě OpenStack s cílem poskytovat různé typy databází, a to uvnitř infrastruktury tak i vně.

V rámci databází umožňuje spustit různé databázové servery s rozdílnými typy topologií. Relační i ne-relační databázové struktury lze využít v rámci infrastruktury jako zdroje.

Modul Trove je závislý na modulech Cinder, Keystone, Glance, Nova a Swift.

[1], [2], [7]

2.1.24 Masakari

Specializovaný modul, designovaný pro zajištění vysoké dostupnosti běžících instancí serverů. Tento modul sleduje a v případě chyby automaticky znovu spouští instance serverů, které byly neočekávaně ukončeny interní nebo externí neplánovanou činností. Při výpočtech může dojít k různým typům chyb, které jsou zapříčiněny například špatnými výpočty, chybnou konfigurací operačních systémů virtuálních serverů nebo například i hardwarovými poruchami infrastruktury, na které je spuštěna platforma OpenStack. Tyto chyby s sebou nesou riziko sníženého výkonu nebo kompletního selhání běžících výpočtů nebo služeb.

Modul Masakari umí zotavit selhané instance virtuálních serverů, které byly spuštěny na bázi softwaru KVM a přes poskytovanou API umožňuje sledovat stav zotavení a konfigurovat parametry, kdy a jak má k zotavení dojít.

Modul Masakari je závislý na modulech Keystone a Nova.

[1], [2], [7]

2.1.25 Murano

Aplikační katalog v rámci infrastruktury poskytuje modul Murano. Tento modul umožňuje vývojářům a administrátorům publikovat rozmanité aplikace, nejčastěji předem připravené pro cloudové nasazení, v rámci sjednoceného aplikačního katalogu.

Pomocí takového katalogu lze velmi zjednodušit vyhledávání, filtrování a spuštění těchto aplikací pomocí sjednoceného rozhraní a ovládacím prvkům. Pro funkci tohoto katalogu je na pozadí této služby použit společně s modulem Murano také modul Heat, který slouží k obsluze a orchestraci zdrojů asociovaných s vybranou aplikací.

Modul Murano je závislý na modulech Keystone a Heat.

[1], [2], [7]

2.1.26 Solum

Solum je modul starající se o automatizaci životního cyklu aplikací a jejich vývoje. Typicky lze tento modul použít pro zjednodušení vývoje aplikací, kde výsledkem těchto aplikací má být sestavený kontejner či archív, na který je daná služba nasazena či přímo spuštěna.

Pro správu používá modul Solum balíky, tzv. „languagepack“, které lze poté použít na vývoj a správu aplikací. Aplikace se dodatečně konfiguruje v souborech formátu YAML.

Modul Solum je závislý na modulech Keystone a Heat.

[1], [2], [7]

2.1.27 Freezer

Pro případy, kdy může v infrastruktuře vzniknout katastrofální selhání systému nebo ztráta dat lze využít modul Freezer. Cílem je vytvářet, distribuovat a obnovovat zálohy z různých míst, operačních systémů a hardwaru.

Modul Freezer umožňuje vytvářet blokové zálohy, které lze poté distribuovat na různá místa, připojená k infrastruktuře. Takovéto zálohování je nezávislé na operačních systémech k operačním systémům, modul Freezer lze spustit na mnoho jiných distribucích a typech operačního systému, zejména pak i na operačních systémech typu Microsoft Windows, Linux, BSD a Apple macOS.

Nejčastější typy záloh, které modul Freezer také podporuje, jsou následující:

- souborové zálohy,
- inkrementální zálohy,
- časové zálohy,
- synchronizační zálohy.

Modul Freezer je závislý na modulu Keystone.

[1], [2], [7]

2.1.28 EC2API

Modul, sloužící jako proxy, pro připojení k instancím EC2 běžících v cloudovém prostředí AWS, poskytovaném společností Amazon.

Cílem je přemostit rozdíly mezi instancemi EC2 v cloudu AWS a instancemi modulu Nova. V případech, kdy je žádoucí převést běh virtuálního serveru do cloudového prostředí AWS, lze tento modul použít pro tento úkol.

Modul EC2API je závislý na modulech Keystone, Nova, Cinder, Neutron a Glance.

[1], [2], [7]

2.1.29 Horizon

Pro vizualizaci stavu kompletní infrastruktury slouží modul Horizon. Jedná se o modul poskytující nástěnku, dostupnou přes webové rozhraní a přístupné pomocí webového prohlížeče. Kompatibilita je zajištěna a otestována s moderními prohlížeči.

Klíčovou závislostí je modul Keystone, pomocí kterého autorizuje modul Horizon jednotlivé uživatele a administrátory, kteří mají přístup k nástěnce.

Komplexnější popis, instalace, konfigurace a použití se nachází v kapitole 8.

Modul Horizon je závislý na modulu Keystone.

[1], [2], [7]

2.1.30 Skyline

Modul Skyline je další implementací nástěnky platformy OpenStack. Jedná se o modernější a rychlejší ekvivalent modulu Horizon s cílem vylepšení nástěnky pomocí moderních technologií a se zaměřením na optimalizace uživatelského rozhraní a zjednodušení používání.

Modul Skyline, stejně jako modul Horizon je závislý na modulu Keystone.

[1], [2], [7]

2.2 Operační moduly

Pro správnou funkci infrastruktury a platformy OpenStack jsou nutné moduly pro řízení dané platformy. K tomuto účelu slouží operační moduly, které usnadňují administrátorům a technikům, kteří nasazují platformu, práci s platformou.

[1]

2.2.1 Ceilometer

Modul s cílem efektivního sběru dat z platformy OpenStack. Zároveň se stará i o jejich normalizaci a transformaci pro vytvoření různých pohledů na data, se kterými se pracuje. Zároveň modul Ceilometer buduje telemetrii pro nástěnku, kde jsou veškeré tyto data k dispozici.

Typicky lze využít modul Ceilometer pro následující úlohy:

- zpoplatnění služeb,
- sledování prostředků a zdrojů,
- nastavení alarmů.

Modul Ceilometer není závislý na jiných modulech.

[1], [2], [7]

2.2.2 Monasca

Modul Monasca poskytuje multi-tenantní, vysoce škálovatelnou a výkonnou službu monitorování (MaaS, monitoring-as-a-service). Pro přístup k monitorování a záznamům metrik poskytuje REST API. Dodatečně poskytuje modul Monasca také i notifikace a upozornění, založené nad zpracováním proudových dat.

Pro komplexní monitorování napříč celou infrastrukturou lze v modulu Monasca nasadit větší množství instancí v mikro-servisní architektuře. Pro vnitřní řízení modulu Monasca napříč infrastrukturou je mezi jednotlivými instancemi použit framework Apache Kafka, jenž slouží k rozesílání zpráv určitým pod-modulům či databázím.

Modul Monasca není závislý na jiných modulech.

[1], [2], [7]

2.2.3 Watcher

Modul Watcher slouží k sledování výkonu a následným optimalizacím běžících instancí virtuálních serverů. Tento modul obsahuje API pro příjem metrik výkonu, optimalizací procesů a plánování akcí a zároveň také poskytuje nástěnku integrovatelnou s modulem Horizon. Správně použití modulu může zlepšit výkon infrastruktury a zároveň s tím i snížit náklady.

Modul Watcher není závislý na jiných modulech.

[1], [2], [7]

2.2.4 Vitrage

Specializovaný modul na sledování příčin událostí a jejich shromažďování. Modul Vitrage slouží také k jejich organizaci, seskupování a vizualizaci, které nám mohou pomoci snáze diagnostikovat příčiny problémů a událostí, které mohou s takovými problémy přímo i nepřímo souviset.

Funkcionalitu modulu Vitrage lze popsat následujícími body:

- mapování fyzických entit na virtuální,
- dedukce stavů, alarmů a událostí,
- hledání a analýza kořene problémů,
- vizualizace těchto faktů v modulu Horizon.

Modul Vitrage není závislý na jiných modulech.

[1], [2], [7]

2.2.5 Adjutant

Modul Adjutant slouží k automatizaci a poskytování API pro organizování automatizovaných byznysových procesů, které přímo souvisí s účty v rámci infrastruktury i mimo ni. Tento modul má doplnit chybějící funkcionalitu modulu Keystone, která by v rámci modulu Keystone nebyla žádoucí z důvodu rozšíření funkcionality nad rámec vymezených hranic modulu.

Cílem tohoto modulu je zjednodušit administrátorské úkony v rámci platformy OpenStack. Tyto procesy jsou založeny na třech krocích:

1. prvotní podání požadavku,
2. schválení požadavku administrátorem nebo pověřenou osobou,
3. odeslání tokenu, potvrzující dokončení operace.

Modul Adjutant není závislý na jiných modulech.

[1], [2], [7]

2.2.6 Cloudkitty

Modul primárně zaměřený na sběr metrik a jejich překládání na finančně spočitatelné prostředky, které mohou být následně předány k fakturaci. Cílem je poskytovat službu RaaS (rating-as-a-service).

Pro konfiguraci tohoto modulu a správného výpočtu využití prostředků, na kterých jsou zakládány finanční metriky, se používají buďto tabulky s předem připravenými cenami, pevně svázanými s jednotlivými metrikami nebo vlastní skripty, které lze nakonfigurovat pro specifické potřeby nabídky a využití infrastruktury. K tomu slouží dostupná API, ke které se přistupuje na základě rozsahů a metrik.

Tento modul nenabízí samotné výpočty pro finanční nacenění produktu nebo služeb, nýbrž poskytuje pouze sledování metrik do spočitatelných tvarů.

Modul Cloudkitty není závislý na jiných modulech.

[1], [2], [7]

2.2.7 Rally

Modul zabývající se sledováním výkonu a jeho testováním, například při nasazení nového softwaru nebo služby. Tento modul slouží k automatizaci sledování, škálování a porovnávání výkonu s předem určenou normou.

Tento modul lze využít jako základní formu nástrojů CI/CD (continuous integration/continuous deployment). Cílem je zautomatizovat sledování prostředků a profilování zdrojů v rámci změny běžícího kódu v operačním systému, detekování problémů se škálováním, výkonem a následnou analýzou hardwaru, která umožní odladění na požadované výkonnostní požadavky.

Modul Rally není závislý na jiných modulech.

[1], [2], [7]

2.2.8 Tempest

Modul Tempest se zabývá nasazením a spouštěním testů pro nasazované programy, které poběží v platformě OpenStack. Konkrétně přistupuje k dostupné API platformy OpenStack aby mohl modul Tempest monitorovat, validovat a spouštět možné scénáře, se kterými musí běžící programy počítat.

Modul Tempest lze spustit oproti clusteru jednoho až tisícům virtuálních serverů v rámci infrastruktury. Tyto testy slouží k otestování funkcionality běžících serverů a programů, nikoliv k testování výkonu (viz modul Rally).

Modul Tempest je závislý na modulu Keystone.

[1], [2], [7]

2.3 Doplnky ke modulům

Jedná se o doplňky k již běžícím nebo nasazovaným modulům. Typicky ve formě „přídavných modulů“.

[1]

2.3.1 Storlets

Doplňkový modul, sloužící jako rozšíření modulu Swift. Tento modul se stará o schopnosti spouštět uživatelské spustitelné výpočty, které jsou prováděny na uzlech, běžně dedikovaných jako úložná pole.

Tyto aplikace jsou zabalený a zkompileovaný kód (například spustitelný archiv jazyku Java s příponou .jar), který je následně uložen do objektového úložiště modulu Swift. Tyto Storlets archívy mohou být poté spuštěny na vyžádání, přímo na úložném uzlu, na kterém se nacházejí.

Oproti předchozím modulům se minimální hardwarové nároky na spuštění uzlu, na kterém poběží modul Storlets liší od dříve vyvozených požadavků, sepsaných v kapitole č. 1.1.1. Minimální požadavky na uzel jsou operační systém Ubuntu Server 14.04 s diskovým prostorem alespoň 10 GB s doporučenou instalací na nahraditelného virtuálního klienta.

Název modulu Storlets vyplývá z běžného používání přípony „let“ (v množném čísle „lets“). Tato přípona typicky udává spustitelnou část kódu, která je typicky asociována s předponou. Běžně se v jazyce Java používaly tzv. applety (aplikace) nebo servlety (serverová aplikace). Název modulu Storlets vznikl tedy jako spustitelná aplikace na úložném médiu (v angličtině storage), konkrétně slova storage a přípony „lets“.

Modul Storlets je závislý na modulu Swift.

[1], [2], [7]

2.4 Integrační nástroje

Tento software, resp. softwarové nástroje, poskytuje integraci v rámci infrastruktury, na které není spuštěna výhradně platforma OpenStack.

[1]

2.4.1 Kuryr

Moduly běžící v kontejnerové infrastruktuře nemusí mít dostatečné připojení k jiným, heterogenním systémům, ať virtualizačním tak i těm běžícím přímo v platformě OpenStack. Řešením toho problému je implementace modulu Kuryr.

Modul Kuryr má za cíl blíže integrovat síťovou komponentu v rámci platformy OpenStack a abstrahovat tak podstatnou část síťové infrastruktury tak, aby bylo možné zprovoznit komunikaci v rámci platformy i pro jiné, rozdílné virtualizační nástroje či běžící kontejnery.

Modul Kuryr je závislý na modulu Neutron.

[1], [2], [7]

2.4.2 Tacker

Modul Tacker slouží jako orchestrátor komunikace pro běh nad heterogenním či proprietárním hardwarem. Dosahuje tohoto cíle pomocí abstrakce založené na principu NFV.

Princip NFV je oddělení síťové infrastruktury od přímého hardwaru a jejím sestavením v rámci softwarové architektury. Od principů softwarově definovaných sítí se liší v principu běhu a abstrakci. Zatímco SDN čistě nahrává chování na specifický hardware, modul Tacker virtualizuje veškerá síťová rozhraní a protokoly, aby nedošlo k možným problémům při komunikaci s rozmanitým nebo proprietárním hardwarem.

Modul Tacker není závislý na žádném modulu.

[1], [2], [7], [13], [14]

3 KLIENSKÉ NÁSTROJE

Nástroje sloužící k připojení k platformě OpenStack a její správě. Dodatečně pod tuto kategorii spadá také softwarový vývojový balík pro automatizaci klientských aplikací.

[1]

3.1 CLI

Zkratka CLI je zkratkou pro ovládací rozhraní příkazové řádky. Takovéto nástroje slouží k ovládání a čtení údajů ohledně platformy OpenStack v terminálu nebo příkazové řádce, tedy bez grafického uživatelského rozhraní.

3.1.1 OpenStackClient

OpenStackClient je jediným, oficiálním nástrojem, poskytující sjednocenou kontrolní vrstvu nad částmi platformy OpenStack. Pro správu všech dostupných modulů běžících v infrastruktuře pracuje OpenStackClient se sjednocenými příkazy.

Mezi dostupné moduly, které lze ovládat pomocí aplikace OpenStackClient jsou moduly:

- Nova,
- Keystone,
- Glance,
- Swift,
- Cinder.

3.1.2 SDK

Zkratka pro Software Development Kit, balík vývojových nástrojů, který umožňuje vyvíjet aplikace s rozšířenou působností. Typicky se nejedná se o samostatně použitelné nástroje, ale pouze o sadu knihoven pro práci s daným softwarem.

[1]

3.1.3 OpenStackSDK

OpenStackSDK je vývojový balík pro jazyk Python, který zprostředkovává pomocí knihovnických funkcí přístup k REST API. Primárně slouží ke komunikaci s platformou OpenStack za pomoci naprogramovaných aplikací.

[1], [2], [7]

4 NÁSTROJE PRO NASAZENÍ

Platforma OpenStack se dá nasadit do infrastruktury různými způsoby, ať už manuálně nebo automatizovaně. Automatizované nasazení napříč celou infrastrukturou je preferováno z důvodu kratšího času nasazení, menší náchylnosti k chybám a většího souladu.

[1]

4.1.1 OpenStack-Helm

Nasazení platformy OpenStack lze i pomocí softwaru Helm. Samotný software Helm je nadstavbou pro spuštění aplikací nebo operačních systémů v platformě Kubernetes. Pro nástroj Helm jsou vydávány tzv. schémata (v anglické literatuře známé jako „charts“), pomocí kterých je platforma OpenStack nasazena na cluster platformy Kubernetes.

[1], [2], [7]

4.1.2 Kolla-Ansible

Platformu OpenStack lze také nasadit, ačkoliv v kontejnerové podobě, i za pomoci softwarového nástroje Ansible. Skripty pro nástroj Ansible sestaví obraz ovládacího uzlu platformy OpenStack a nasadí ho na specifikovaný server. Cílem je co nejvíce zjednodušit přípravu a nasazení platformy OpenStack a zároveň poskytnout administrátorům možnost rychle nasadit servery, připravené na produkční prostředí.

[1], [2], [7]

4.1.3 Kayobe

Skupina nástrojů a skriptů Kayobe má za cíl nasadit kontejnerizovanou platformu OpenStack přímo na servery. Spolupracuje s nástrojem Bifrost pro zjištění a provizi serverů, pomocí kterých budou sestaveny kontejnery platformy OpenStack.

[1], [2], [7]

4.1.4 OpenStack-Ansible

Sada průvodních poznámek a interaktivních postupů pro nasazení a konfiguraci již existující platformy OpenStack pomocí nástroje Ansible.

[1], [2], [7]

4.1.5 OpenStack-Charms

Nasazení kontejnerizovaného softwaru OpenStack za použití softwarových nástrojů Charms a Juju.

[1], [2], [7]

4.1.6 Bifrost

Průvodce nasazení softwaru OpenStack pomocí služby Ironic.

[1], [2], [7]

4.1.7 OpenStack-Chef

Příručky pro sestavení, nasazení a užívání softwaru OpenStack pomocí softwaru Chef.

[1], [2], [7]

4.2 Sestavovací návody pro populární nástroje

Pro potřeby bližšího ujasnění, jak nasadit platformu OpenStack do jiných prostředí.

[1]

4.2.1 LOCI

Příručky a návody pro sestavení lehkých OCI kontejnerů a obrazů pro služby softwaru OpenStack.

[1], [2], [7]

4.2.2 Puppet-OpenStack

Moduly softwaru Puppet pro škálovatelné a automatizované nasazení softwaru OpenStack do cloudových platforem.

[1], [2], [7]

Praktická část

V následujících kapitolách, je detailně popsán postup instalace softwaru OpenStack, konkrétně v provedení DevStack, a jeho základní konfigurace pro zprovoznění. Pro dodatečné konfigurační parametry by měl uživatel jako referenci použít oficiální dokumentaci dané služby.

[1], [2], [7]

5 PRVOTNÍ NASTAVENÍ PROSTŘEDÍ

Vývoj softwaru OpenStack je realizován v operačním systému Ubuntu v serverové verzi a je založen na operačním systému Debian. Jedná se o jednu z oficiálně podporovaných distribucí, které mají k dispozici software OpenStack ve svých repozitářích, a tak není nutné hledat dodatečné repozitáře nebo stahovat kód dodatečně, mimo balíčkovací software.

[15], [16]

5.1 Instalace operačního systému

Pro instalaci operačního systému je nutné sehnat obraz instalačního disku nebo jiného alternativního instalačního média. Operační systém Ubuntu Server má integrovaný instalátor, pomocí kterého se operační systém instaluje interaktivně kde k instalaci postačí pouhá klávesnice.

[15], [16]

5.1.1 Stažení obrazu ISO

Obraz instalačního disku, distribuovaný ve formátu ISO, lze najít na oficiálních webových stránkách operačního systému Ubuntu. Po stažení obrazu je nutné tento obraz nahrát na vybrané instalační médium. Fyzická média pro instalaci jsou následující:

- CD disk (max. 700 MB),
- DVD disk (max. 4.7 GB),
- USB flash disk (nejčastěji v jednotkách až desítkách GB).

Pro vypálení instalačního obrazu na disky typu CD nebo DVD je možné využít dostupný software pro vypalování dat na disky CD a DVD. Pro USB flash disky lze taktéž použít některé z těchto nástrojů, kde většina z takových nástrojů fungují pouze pod operačním systémem Windows. Pro potřeby nahrání obrazu na USB flash disk z operačního systému Windows postačí otevřený software Rufus, který lze stáhnout z webových stránek nebo přímo ze serverů společnosti GitHub.

[19], [20]

Z Linuxových operačních systémů plnohodnotně postačí nástroj `dd`, pomocí kterého nakopírujeme celý instalační obraz na dané zařízení. Tímto příkazem se ale vymažou veškeré uložené soubory a oddíly, které již existují na disku.

```
lukas@home:~# dd status=progress if=~Downloads/ubuntu-server.iso of=/dev/sdf
```

Příkaz 1: Nahrání instalačního obrazu na USB flash disk v Linuxovém systému

[21]

Po úspěšném nahrání či vypálení instalačního obrazu na disk, stačí již daný disk připojit do požadovaného PC nebo serveru a spustit instalaci.

[15], [16], [17]

5.1.2 Instalace operačního systému

Po vložení instalačního disku je nutné zajistit že se vybere právě tento instalační disk a BIOS nezačne spouštět již předinstalovaný systém (pokud je takový systém přítomen). Docílí se toho změnou priorit disků, ze kterých se bude zavádět operační systém. Pro tuto změnu je nutné konzultovat manuál základní desky, která se nachází v počítači či serveru (dále jenom serveru).

Po startu serveru, při správném nastavení priorit disků, se začne spouštět instalátor. Podle připojených interních komponentů serveru, periférií anebo i rychlosti média, na kterém se nachází instalační disk, může inicializace tohoto instalátoru trvat až několik minut.

Po dokončení inicializace se na obrazovce objeví průvodce instalací. V tomto průvodci se pomocí klávesnice interaktivně nastavují vyžadované parametry, které jsou nutné pro provedení instalace.

Instalace vyžaduje zadat nebo nakonfigurovat následující údaje:

- jazyk instalace,
- jazyk a rozložení klávesnice,
- nastavení IP adresy u požadovaných rozhraní (výchozí je DHCP),
 - případně vytvoření spojených rozhraní (bond),
- adresu proxy serveru (pokud je vyžadována/existuje),
- archivní URL adresa repozitáře (výchozí je předvyplněna),
- rozložení disku (pro minimální instalaci postačí možnost „využít celý disk“),
- systémové a uživatelské údaje:
 - jméno uživatele,
 - název serveru,

- uživatelské jméno,
- a uživatelské heslo.

Pro správu serveru přes síť je nutné zaškrtnout položku „Nainstalovat OpenSSH server“, pro umožnění síťové dostupnosti pomocí protokolu SSH. Poté se zobrazí poslední okno v instalátoru a je zde prezentována nabídka v podobě seznamu instalovatelných komponent. Není nutné v tento moment vybírat žádnou z nich. Postup pro jejich instalaci je shodný jako u serveru SSH, jelikož jsou ale veškeré ostatní prerekvizity v operačním systému Ubuntu Server nainstalovány, jsou tyto komponenty plně volitelné, dle uvážení administrátora.

[15], [16], [17]

5.2 Instalace prerekvizit a příprava uživatele

Před samotnou instalací platformy OpenStack je nejdříve nutné připravit systém pro nasazení této platformy. Následující postup dodržuje doporučení pro instalaci platformy a konvence linuxových operačních systémů.

[22]

5.2.1 Nástroj git

Pro zprovoznění platformy OpenStack je vyžadováno stažení samostatného instalačního skriptu, tak i dodatečných nástrojů, které jsou v instalačním skriptu vyžadovány. Mezi jeden z hlavních nástrojů, pomocí kterého se získají veškeré instalační soubory, se řadí nástroj git.

Nástroj git slouží primárně k distribuovanému verzování kódu, avšak pro potřeby instalace platformy jsou využity příkazy nástroje git pro naklonování repozitáře instalačního skriptu DevStack. Tento nástroj je ve většině případů předinstalován na mnoho populárních distribucích linuxových operačních systémů. Pokud nástroj git není předinstalován, tak lze tento nástroj velmi snadně manuálně doinstalovat, a to za pomoci balíkovacího systému, kterým disponuje téměř každá distribuce.

[22], [23]

Instalaci nástroje git se provádí následujícím příkazem, kde jsou uvedeny různé balíkovací systémy. Pro zjištění, jaký balíčkovací systém používá konkrétní distribuce, se podívejte do dokumentace dané distribuce. Následující balíkovací systémy odpovídají oficiálně certifikovaným operačním systémům, s plnou podporou softwaru OpenStack:


```
# Debian, Ubuntu
lukas@home:~# apt install git

# RHEL nebo CentOS
lukas@home:~# yum install git
```

Příkaz 2: Instalace nástroje git pomocí balíkových nástrojů

[15]

5.2.2 Vytvoření uživatele stack

Dle konvencí moderních víceuživatelských operačních systémů, je vysoce žádoucí oddělit běžné uživatele od administrátora a administrátorských práv. Kvůli tomu je důležité vytvořit uživatele, pod kterým poběží celá infrastruktura (konkrétně na jednom vybraném uzlu).

Všechny certifikované systémy mají dostupnou správu uživatelů, pokud není explicitně zakázána, kde každý uživatel má přiřazené svoje vlastní přihlašovací jméno, případně heslo a dodatečné údaje. Vytvoření uživatele stack, pod kterým platforma OpenStack poběží, se vytváří následovně:

```
lukas@home:~# useradd -s /bin/bash -d /opt/stack -m stack
```

Příkaz 3: Vytvoření samostatného uživatele stack

[22], [24]

Dle doporučení vývojářů platformy OpenStack, je zapotřebí zajistit, že daný adresář, ve kterém se bude nacházet domovský prostor uživatele stack, je dostupný k přepnutí. Tento příkaz zajistí správné nastavení práv na domovský adresář uživatele stack:

```
lukas@home:~# chmod +x /opt/stack
```

Příkaz 4: Přepsání práv domovského adresáře uživatele stack

[22], [25]

Nadále je nutné zajistit že uživatel stack má administrátorská práva. Dle vybrané distribuce operačního systému se mohou způsoby přidání administrátorských práv lišit. Po použití distribuci operačního systému Ubuntu Server lze použít nástroj visudo, který otevře soubor `/etc/sudoers`, do kterého pak může administrátor přidat, změnit či odebrat uživatele či skupiny, které mají mít přístup k právům administrátora, specificky přes příkaz `sudo`. Vyvolání příkazu `visudo`, společně s preferovaným editorem, se provede následovně:

```
lukas@home:~# EDITOR=/usr/bin/nano visudo
```

Příkaz 5: Přidání uživatele stack do souboru `sudoers` s editorem nano

[26], [27]

Po otevření souboru ve vybraném editoru (v ukázce použit editor nano) zavedeme nový záznam do souboru, na samostatný řádek, ve formátu:

```
stack ALL=(ALL) NOPASSWD: ALL
```

Kód 1: Nový záznam uživatele stack v souboru `/etc/sudoers`

[22]

Pokud vše bylo nastaveno správně a uživatel stack existuje, pak pro další používané příkazy bude používán uživatel stack, jak bude názorně uvedeno v příkazových segmentech. Přepnutí se na uživatele stack provedeme za pomoci administrátorského příkazu:

```
lukas@home:~# sudo -u stack -i
stack@home:~$ whoami
stack
stack@home:~$ pwd
/opt/stack
```

Příkaz 6: Přepnutí na uživatele stack

[22], [28], [29], [30]

Výpis příkazu `pwd` by měl odpovídat složce, zadané v příkazu Příkaz 3, pomocí přepínače `-d`. Pokud se složky shodují tak současný uživatel, který je nyní přihlášen, je uživatel stack a nachází se ve své domovské složce.

6 INSTALACE SOFTWARE OPENSTACK

S připraveným prostředím a nainstalovaným systémem lze přejít k samotné instalaci platformy OpenStack.

6.1 Stažení instalačního balíčku

Instalace platformy OpenStack je řízená pomocí příkazové řádky, která je použita jak ke stažení, tak i k její instalaci. Veškerá práce s následujícími příkazy probíhá pod uživatelem stack.

6.1.1 Naklonování repozitáře softwaru OpenStack

Pro instalaci platformy se musí stáhnout prvotně instalační repozitář, obsahující skripty programu DevStack, který umožní nainstalovat a spustit izolovaný kontrolní uzel platformy OpenStack. Iniciální klonování se provede následujícím příkazem:

```
lukas@home:~# sudo -u stack -i
stack@home:~$ git clone https://opendev.org/openstack/devstack
Cloning into 'devstack'...
remote: Enumerating objects: 50660, done.
remote: Counting objects: 100% (30804/30804), done.
remote: Compressing objects: 100% (10278/10278), done.
remote: Total 50660 (delta 30058), reused 20526 (delta 20526), pack-reused 1985
Receiving objects: 100% (50660/50660), 9.44 MiB | 2.28 MiB/s, done.
Resolving deltas: 100% (35983/35983), done.
```

Příkaz 7: Stažení instalačního repozitáře DevStack

[22], [23]

Po dokončení příkazu by měla být dostupná nová složka se staženým repozitářem, pojmenována podle poslední, nejhlubší, složky v cestě stahovaného repozitáře, převedené na malá písmena. Pro potvrzení postačí příkaz `ls`:

```
stack@home:~$ ls -l
total 0
drwxrwxr-x 1 stack stack 520 Mar 29 17:43 devstack
```

Příkaz 8: Kontrola existence složky staženého repozitáře

[22], [31]

6.1.2 Příprava instalace

Před samotnou instalací je zapotřebí ještě vytvořit soubor s nastavenými hesly. Tyto hesla jsou konkrétně čtyři a slouží k připojení k následujícím službám:

- administrace,
- databáze,
- služba RabbitMQ,
- servisní služba.

Hesla musí být uložena v instalačním repozitáři ještě před samotným spuštěním instalace. Pro přepnutí do složky a vytvoření těchto hesel pomocí editoru nano se použijí tyto příkazy:

```
stack@home:~$ cd devstack/  
stack@home:~/devstack$ nano local.conf
```

Příkaz 9: Přepnutí do složky devstack a vytvoření konfigurace hesel

[27], [32]

Po otevření editoru vložíme do souboru `local.conf` následující text ve formátu:

```
[[local|localrc]]  
ADMIN_PASSWORD=UPCEst58238  
DATABASE_PASSWORD=$ADMIN_PASSWORD  
RABBIT_PASSWORD=$ADMIN_PASSWORD  
SERVICE_PASSWORD=$ADMIN_PASSWORD
```

Kód 2: Uložená hesla v souboru `local.conf`

[22]

V tomto formátu textového dokumentu stačí zadat jedno heslo, sdílené mezi službami. Samozřejmostí je možnost zadat 4 různá hesla. Kvůli způsobu naprogramování platformy OpenStack mohou tyto hesla obsahovat pouze alfanumerické znaky, tj. velká a malá písmena anglické abecedy společně s číslicemi. Speciální znaky mohou vést k nefunkčnosti určitých služeb, kvůli přítomnosti speciálních znaků v tomto souboru nebo v heslech.

[22]

6.2 Instalace softwaru

Po naklonování repozitáře a vytvoření souboru `local.conf` s hesly pro služby platformy OpenStack.

6.2.1 Spuštění instalace

Samotná instalace platformy OpenStack se spustí pomocí instalačního skriptu `stack.sh`. Tento skript by měl být spuštěn jako uživatel `stack`, jak bylo psáno v předchozí kapitole.

```
stack@home:~/devstack$ ./stack.sh
```

Příkaz 10: Spuštění instalace platformy OpenStack

[22]

Po spuštění instalace se začnou do terminálu vypisovat řádky s výstupy jednotlivých programů a příkazů, které dávají administrátorovi vědět co všechno se z instalačního skriptu provádí. Vzhledem k rychlostem moderního hardwaru a optimalizací operačních systémů je tento výpis výstupů velmi rychlý a jednotlivé řádky jsou velmi rychle přepsány novými v rámci průběhu instalace.

6.2.2 Průběh a konec instalace

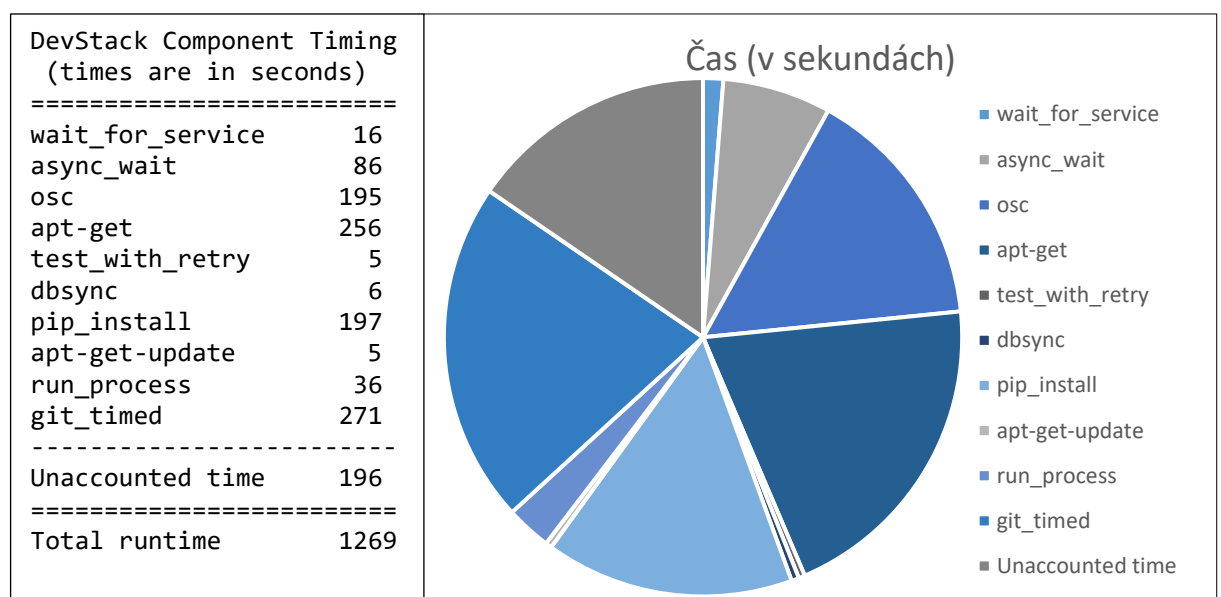
Celá instalace probíhá samostatně, bez zásahu administrátora. I přesto je, ale velmi sině doporučeno sledovat průběh instalace, aby byl při případných chybách administrátor obeznámen s danou situací a případně jak ji dále řešit.

Během instalace se spouští Z výsledných logů lze vyvodit, že instalace spouští následující programy:

- git,
- python3,
- pip3,
- WSGI:
 - keystoneWSGI,
 - nova-apiuWSGI,
 - cinder-apiuWSGI,
 - placementWSGI,

- etcd,
- neutron-server,
- mysqld,
- ovs-vswitchd,
- neutron-ovn-metadataagent.

Po dokončení instalace jsou uživatelé vypsány informace o instalaci, ze kterých lze vyvodit jaké akce proběhly a jak dlouho trvaly.



Kód 3: Výstup sledovaných časování během instalace vč. grafu

Z výsledného výpisu doby trvání instalace je patrné že na použitém serveru probíhala instalace 1269 sekund, což odpovídá 21 minutám a 9 sekundám. Převážnou část instalace zabírá klonování repozitářů (kategorie `git_timed`), stahování dodatečných prerekvizit pomocí balíkovacího systému `apt` (kategorie `apt-get-update` a `apt-get`) a instalace balíků systému `python` (kategorie `pip_install`). Tyto kategorie jsou přímo závislé na rychlosti stahování na straně internetové přípojky a rychlosti serverů, ze kterých se balíky stahují.

Další položkou ve výsledném výpisu je tabulka provedených databázových dotazů. Z předchozího výstupu lze vyčíst že databázové operace (kategorie `dbsync`) jsou velmi efektivní i přes jejich vysoký počet. Následující tabulka popisuje instalované moduly a operace, které vyvolali nad databázovým serverem.

Tabulka 2: Počet databázových dotazů provedených jednotlivými moduly během instalace

	SELECT	INSERT	UPDATE	DELETE
keystone	35351	93		
neutron	5949	13977	139	27
placement	90	70	3	
nova_api	98	10		
nova_cell_0	362	20	22	
nova_cell_1	458	8	85	
cinder	245	5	1	1
glance	107	54	2	

Poslední položkou vypsanou po instalaci, před návratem do příkazové řádky, je sekce popisující komponenty, adresy IP, verze operačního systému, informace o verzi platformy OpenStack a způsob běhu nainstalovaných služeb.

```
This is your host IP address: 10.110.0.56
This is your host IPv6 address: ::1
Horizon is now available at http://10.110.0.56/dashboard
Keystone is serving at http://10.110.0.56/identity/
The default users are: admin and demo
The password: UPCEst58238

Services are running under systemd unit files.
For more information see:
https://docs.openstack.org/devstack/latest/systemd.html

DevStack Version: 2024.1
Change: f4f09416ca8de09820b52599c7bcd2911b407466 Merge "Drop unused environments
for Triple0 and heat agents" 2024-03-19 18:55:14 +0000
OS Version: Ubuntu 22.04 jammy

2024-03-30 18:36:32.960 | stack.sh completed in 1269 seconds.
```

Kód 4: Finální výstup instalačních informací po dokončení instalace

Z vypsáných informací lze vyčíst že platforma OpenStack byla úspěšně nainstalována, uživatelé admin a demo jsou dostupní se zadaným heslem, které bylo specifikováno v souboru `local.conf`. Adresa IP, na které naslouchá primární uzel platformy OpenStack je 10.110.0.56 pro protokol IPv4 a ::1, značící dostupnost pouze přes adresu tzv. localhost, protokolu IPv6 (Toto je způsobeno vypnutým protokolem IPv6 na připojených rozhraní). Pokud by byla nastavena adresa IPv6 na připojeném rozhraní tak by tento uzel naslouchal i na adrese IPv6.

6.2.3 Sledování prostředků během instalace a běhu platformy

Během instalace je možné z dalšího terminálu sledovat vytížení prostředků, spuštěné procesy, vytížení disků, sítě anebo CPU. Po navázání nového připojení v odděleném terminálu, například přes protokol SSH, lze spustit rozdílné nástroje, které umožní sledovat vytížení prostředků.

Jedním z těchto programů je aplikace top, případně její variace jako jsou atop, htop, btop (někdy označován jako bashtop) a další, které slouží k obecnému monitorování procesů, v jakémkoliv stavu. Různé varianty jsou obohaceny o dodatečné sledované prostředky, například vytížení nebo rychlost sítě, zatížení a rychlost čtení a zápisu na diskové oddíly nebo dodatečně vypisují do konzole blízkou historii těchto využití v různých podobách.

```
 0[] 3[] 6[] 9[] 12[] 15[] 18[] 21[] 24[] 27[] 30[] 33[] 36[] 39[] 42[] 45[]
 1[] 4[] 7[] 10[] 13[] 16[] 19[] 22[] 25[] 28[] 31[] 34[] 37[] 40[] 43[] 46[]
 2[] 5[] 8[] 11[] 14[] 17[] 20[] 23[] 26[] 29[] 32[] 35[] 38[] 41[] 44[] 47[]
Mem[|||||] 13.06/62.86 Tasks: 241, 1020 thr; 2 running
Swp[ ] 0K/12.0G Load average: 0.09 0.11 0.18
Uptime: 03:28:54

  PID USER      PRI  NI  VIRT   RES   SHR  S  CPU% MEM%  TIME+  Command
142229 root        20   0 45456 31700 15768 R 13.6  0.0  0:00.21 /opt/stack/data/venv/bin/python3.10 /opt/stack/data/v
142216 lukas       20   0 10048  5768  3444 R  3.2  0.0  0:00.52 htop
 17753 root        20   0 12252  6988  4712 S  1.9  0.0  3:33.75 /var/lib/pcp/pmdas/linux/pmdalinux
112002 stack       20   0 153M   106M  7620 S  1.9  0.2  0:11.02 /opt/stack/data/venv/bin/python3.10 /opt/stack/data/v
96328 root        10  -10 4479M  512M 12520 S  1.3  0.8  1:55.95 ovs-vswitchd unix:/var/run/openvswitch/db.sock -vcons
20381 root        19  -1 265M   177M 174M S  0.6  0.3  0:18.80 /lib/systemd/systemd-journald
22077 rabbitmq    20   0 8566M 169M 64120 S  0.6  0.3  0:02.60 /usr/lib/erlang/erts-12.2.1/bin/beam.smp -W w -MBas a
62787 mysql      20   0 25.4G 1258M 39396 S  0.6  2.0  1:46.56 /usr/sbin/mysqld
63672 stack     20   0 31544 21084 9228 S  0.6  0.0  1:18.49 python3 /bin/dstat -tcmdrylpg --tcp --output /opt/st
63673 stack     20   0 31544 21124 9244 S  0.6  0.0  0:55.26 python3 /bin/dstat -tcmdrylpg --swap --tcp
96382 root        10  -10 4479M  512M 12520 S  0.6  0.8  0:08.52 ovs-vswitchd unix:/var/run/openvswitch/db.sock -vcons
97016 stack     20   0 153M   126M 19860 S  0.6  0.2  0:04.93 neutron-ovn-metadata-agent (/opt/stack/data/venv/bin/
106156 stack     20   0 138M   116M 19960 S  0.6  0.2  0:30.67 /opt/stack/data/venv/bin/python3.10 /opt/stack/data/v
106902 stack     20   0 139M   105M 7916 S  0.6  0.2  0:01.79 /opt/stack/data/venv/bin/python3.10 /opt/stack/data/v
F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice F8Nice +F9Kill F10Quit
```

Obrázek 1: Příklad běžícího programu htop v terminálu, po instalaci platformy OpenStack

Pro monitorování přímého využití disků či dalších paměťových médií slouží například aplikace iotop, která v nových verzích jádra operačních systémů Linux vyžadují administrátorské oprávnění pro monitorování procesů a jejich zápisů a čtení z paměťových médií. Tento program ve výchozím nastavení shrnuje celkovou rychlost přístupu k paměťovým médiím, rozdělenou na čtení a zápis.

6.2.4 Dodatečná konfigurace úložiště

Během zpracování práce bylo zjištěno že pokud neexistuje na zařízení specificky pojmenované úložiště typu LVM, tak nepůjde nijak vytvořit jakýkoliv virtuální diskový oddíl pro práci s virtuálními severy, protože pro vytvoření instancí je zapotřebí mít na systému oddíl, specificky pojmenovaný jako cinder-volumes. Bez tohoto úložiště nelze vytvořit jakýkoliv

diskový oddíl, ze kterého by šel poté vytvořit virtuální server. Vytvoření tohoto oddílu, konkrétně na celém disku `/dev/sdc`, se provede pomocí následujících příkazů:

```
lukas@home:~$ sudo pvcreate /dev/sdc
Physical volume "/dev/sdc" successfully created.
lukas@home:~$ sudo vgcreate cinder-volumes /dev/sdc
Volume group "cinder-volumes" successfully created
lukas@home:~$ sudo pvdisplay
--- Physical volume ---
PV Name                /dev/sdc
VG Name                cinder-volumes
PV Size                931.00 GiB / not usable 4.00 MiB
Allocatable            yes
PE Size                4.00 MiB
Total PE               238335
Free PE                238335
Allocated PE           0
PV UUID                wrCtba-woLr-ZPaV-XwrL-tIH6-q8Kf-7X2Xwb
```

Příkaz 11: Vytvoření nového disku LVM s oddílem cinder-volumes

Ve své podstatě se jedná o nedostatek instalačního skriptu DevStack, kde není předem zkontrolována existence jakéhokoliv předem podporovaného úložiště, specificky pro modul Cinder. Konfigurací modulu Cinder by šlo upravit jaké úložiště se má vybrat a kde takové úložiště hledat, nicméně se jedná o modul nainstalovaný v rámci instalačního skriptu DevStack, kde běžný uživatel předpokládá že vytvořený uzel platformy OpenStack bude disponovat přímo připraveným systémem, bez dalších rozšířených problémů.

7 KEYSTONE

Modul Keystone je jeden ze základních modulů celé platformy OpenStack. Zařizuje přístup k platformě, kompletně obsluhuje autentizaci, autorizaci a stará se o řízení přístupu jednotlivých uživatelů.

7.1 Instalace

Základní instalace je provede automaticky pomocí instalačního skriptu DevStack. Mimo skripty DevStack lze nainstalovat modul Keystone samostatně. K tomu je zapotřebí provést následující akce:

- vytvoření MySQL databáze pro modul Keystone,
- stažení a instalace balíku keystone,
- konfigurace příslušných souborů,
- konfigurace Apache HTTP serveru.

Instalace této služby jako samostatného celku je mimo rozsah této práce a slouží jako služba pro autentifikaci a autorizaci pro produkční infrastrukturu, obsahující typicky již vytvořenou část platformy OpenStack, kde je žádoucí připojit další instanci modulu Keystone, pro snížení zátěže a režie, pokud je autentizační služba vytížena nebo pro urychlení odezvy v rámci velmi rozsáhlé infrastruktury, kde by mohla odezva od centrálního prvku způsobit nepříjemnou odezvu.

[35]

7.2 Spuštění

Dle jednotlivých distribucí operačního systému Linux se služby, běžící na serveru, spouští různými způsoby. Pro použitou distribuci Ubuntu Server verze 22.04 (společně i s distribucemi operačního systému RHEL) se jedná o systém systemd, který se stará o spuštění služeb platformy OpenStack, ihned po spuštění operačního systému.

[35]

7.2.1 Otestování běhu

Pro ujištění, že modul Keystone běží a odpovídá na dotazy, lze provést jednoduchý kontrolní HTTP požadavek z lokálního systému. Tento dotaz by měl vrátit základní informace o běžícím modulu Keystone. Tento dotaz lze provést pomocí příkazu curl, který pošle požadavek HTTP

na zadanou adresu. V tomto případě adresa IPv4, kterou ve finálním výstupu vypsal instalační skript DevStack (v práci konkrétně adresa 10.110.0.56).

```
stack@home:~/devstack$ curl http://10.110.0.56/identity
{"versions": {"values": [{"id": "v3.14", "status": "stable", "updated": "2020-04-07T00:00:00Z", "links": [{"rel": "self", "href": "http://10.110.0.56/identity/v3/"}], "media-types": [{"base": "application/json", "type": "application/vnd.openstack.identity-v3+json"}]}}}
```

Příkaz 12: Příkaz požadavku HTTP na spuštěný modul Keystone

Z výstupu požadavku lze vyčíst že modul Keystone běží a odpovídá na specifikované adrese IPv4. Společně se ve výstupu požadavku vypsala i současně běžící verze modulu Keystone, konkrétně verze 3.14 z větve stable (běžné stabilní vydání modulu Keystone).

7.2.2 Parametry konfigurace

Pro práci s příkazovou řádkou a příkazem `openstack` je nutné zavést do proměnné prostředí, resp. terminálu, ve kterém bude spuštěn příkaz `openstack`. Jedná se o tyto proměnné:

```
OS_USERNAME=admin
OS_USER_DOMAIN_NAME=Default
OS_PASSWORD=UPCEst58238
OS_PROJECT_NAME=admin
OS_PROJECT_DOMAIN_NAME=Default
OS_AUTH_URL=http://localhost/identity/v3 # Nebo http://10.110.0.56/identity/v3
OS_IDENTITY_API_VERSION=3
```

Kód 5: Vyžadované proměnné prostředí pro práci s příkazem `openstack`

Případně je možné zapisovat tyto proměnné jak do samostatného souboru či skriptu, který lze spustit manuálně nebo zavést tyto údaje do souboru `.bash_profile`, který se postará o jejich naplnění při přihlášení uživatele. Pro automatizaci lze zadat tyto řádky do souboru `.bash_profile` pomocí zvoleného editoru:

```
#!/usr/bin/bash
export OS_USERNAME=admin
export OS_USER_DOMAIN_NAME=Default
export OS_PASSWORD=UPCEst58238
export OS_PROJECT_NAME=admin
export OS_PROJECT_DOMAIN_NAME=Default
export OS_AUTH_URL=http://localhost/identity/v3
export OS_IDENTITY_API_VERSION=3
```

Kód 6: Automatizace proměnných pomocí souboru `.bash_profile`

[37], [38]

Po zavedení těchto proměnných prostředí do souboru `.bash_profile`, aby vzešly tyto změny v efekt, je žádoucí buďto načíst tyto nové hodnoty pomocí příkazu `source`:

```
stack@home:~/devstack$ source .bash_profile
```

Příkaz 13: Načtení nových hodnot prostředí pomocí příkazu `source`

[37]

Nebo pomocí odhlášení a opětovného přihlášení uživatele `stack`:

```
stack@home:~/devstack$ exit
lukas@home:~$ sudo -u stack -i
```

Příkaz 14: Načtení nových proměnných pomocí opětovného přihlášení

[28]

Po nastavení těchto proměnných si je lze zkontrolovat pomocí příkazu `env`, kde by měly být vypsaný mezi ostatními proměnnými prostředí. Po ověření, že jsou tyto proměnné prostředí přítomny, lze používat příkaz `openstack` bez dalších prepínačů, které by sloužily ke konfiguraci autentizace. Pro ujištění že dané parametry fungují lze vyvolat příkaz `openstack`, který vytvoří autentizační token, pro použití v jiných scénářích.

```
stack@home:~$ openstack token issue
+-----+-----+
| Field      | Value                                                                 |
+-----+-----+
| expires    | 2024-03-31T20:00:52+0000                                             |
| id         | gAAAAABmCbLk6yx2DLwbB_eH45SpTOjAw8LH1jX0jfbRpdRWTXZHD8JBKrXjXW   |
|            | ZOaDs8FVHiozxiPgW0MeP0mL_Mvy3wMca-h1dGSH5DT7314ehZUE4bSZ-       |
|            | iB8V7wzMCrvIeIIDP6lFE6wrOSQ4UB-                                   |
|            | eme8ecJ2eFITPBFdvXWzca4yPYjDprVY                                   |
| project_id | 419487a4b4f343418c58283f4c83df75                                     |
| user_id    | 7b4f465f4e0548a6b4e58bec461e8b91                                   |
+-----+-----+
```

Příkaz 15: Test příkazu `openstack` s nastavenými proměnnými prostředí

[40]

7.3 Správa projektů, uživatelů a rolí

Základním cílem modulu Keystone je správa projektů, uživatelů a rolí kde veškeré tyto údaje jsou poté propagovány napříč infrastrukturou.

Projekty jsou jednotky nula či více uživatelů, které mají přiřazené určité prostředky, kde například projekt v modulu Nova vlastní určité virtuální servery, zatímco v modulu Zun vlastní projekt specifické kontejnery.

Uživatelé jsou jednotlivé objekty reprezentující typicky jednu danou osobu, která je zařazena do určitých projektů, kde v rámci těchto projektů mají specifické role k daným projektům.

Role jsou přiřazené oprávnění pro jednotlivé uživatele, již se vážou k projektům. Například, uživatel může zastávat roli administrátora v projektu A zatímco jenom běžného obsluhujícího projektu B.

Modul Keystone má potřebné programy a infrastrukturu, aby tyto množiny projektů, uživatelů a rolí efektivně zpracovával a centralizovaně udržoval v necentralizované infrastruktuře.

7.3.1 Správa projektů

Před jakoukoliv manipulací s projekty je vhodné si nejdříve vypsát již vytvořené projekty.

```
stack@home:~$ openstack project list
+-----+-----+
| ID                | Name                |
+-----+-----+
| 2543b057601a49bcbd5816842c177dd7 | alt_demo            |
| 3dcf58d392af4539a22a0ead6920aff8 | invisible_to_admin  |
| 419487a4b4f343418c58283f4c83df75 | admin               |
| cb1503089a804e7d802666ba114e6d28 | demo                |
| d23a5c1c799e4064bbfe63c3039e05e6 | service             |
+-----+-----+
```

Příkaz 16: Test příkazu openstack s nastavenými proměnnými prostředí

[41]

Při kontrole že daný projekt ještě neexistuje se vytvoření nového projektu provede následovně:

```
stack@home:~$ openstack project create --description 'new proj' new-project \
--domain default
+-----+
| Field      | Value                                     |
+-----+
| description | new proj                                 |
| domain_id   | default                                  |
| enabled     | True                                     |
| id          | 885fd62b1ae04dd4b6f6e772bf562f7e       |
| is_domain   | False                                    |
| name        | new-project                              |
| options     | {}                                       |
| parent_id   | default                                  |
| tags        | []                                       |
+-----+
```

Příkaz 17: Založení nového projektu v modulu Keystone

[41]

Z výstupu lze vyčíst údaje o nově vytvořeném projektu. Hlavním identifikátorem, který je použit dále je pole id, které slouží pro manipulaci s projekty. Stejný výstup jako po vytvoření nového projektu lze získat příkazem:

```
stack@home:~$ openstack project show 885fd62b1ae04dd4b6f6e772bf562f7e
```

Příkaz 18: Vypsání vlastností projektu s použitím projektového id

[41]

Úprava projektu, resp. jeho vlastností se provede příkazem:

```
stack@home:~$ openstack project show 885fd62b1ae04dd4b6f6e772bf562f7e
```

Příkaz 19: Změna vlastnosti projektu s použitím projektového id a nového jména

[41]

Poslední možnou operací nad projekty je možnost jejich vymazání, které se provede spuštěním příkazu:

```
stack@home:~$ openstack project show 885fd62b1ae04dd4b6f6e772bf562f7e
```

Příkaz 20: Vymazání projektu s použitím projektového id

[41]

7.3.2 Správa uživatelů

Zbytek příkazů, které slouží ke správě uživatelů, jsou velice podobné těm, které jsou použity pro správu projektů. Příkaz pro výpis uživatelů vypadá následovně:

```
stack@home:~$ openstack user list
```

ID	Name
7b4f465f4e0548a6b4e58bec461e8b91	admin
f8860339929d4201b007e052a0e46e0f	demo
93490487d4a046cdb9365f627f927d8b	demo_reader
87c8b71a0a40438fbc1612641a2bde8e	alt_demo
01b0cb94225a4f58851c22d1f0831a7f	alt_demo_member
c43c862bd198472d85c680e2f419fc5d	alt_demo_reader
9486a6eb05474d699b9c75639b4e182b	system_member
974414d5b66843889b9e43965b021091	system_reader
7efac38e82bd4569a162c546827de1b4	nova
ada297ec056349039562c70c183a39e8	glance
08ddefd9a24b4ee38889cf48a9d559db	cinder
ae5ebc48f1ba4b3b9acc06150fc210a0	neutron
eede92880eea4bc3aa486ad596a0a14f	placement

Příkaz 21: Příkaz openstack pro výpis vytvořených uživatelů

[41]

K vytvoření nového uživatele postačí pouze přihlašovací jméno uživatele, nicméně je, ale žádoucí přidat k tvorbě uživatele rovnou i jeho heslo a ID nebo jméno projektu, ke kterému bude mít nový uživatel přístup. Bez těchto dodatečných informací se nový uživatel nebude moci přihlásit k webové nástěnce modulu Horizon, který je probrán v následující kapitole č. 8. Vytvoření nového uživatele, včetně dodatečných informací se provede následujícím příkazem:

```
stack@home:~ $ openstack user create --project new-project --password new-user  
USER_NAME
```

Příkaz 22: Vytvoření nového uživatele

[41]

Uživatele lze dočasně vypnout a opětovně zapnout, pro efektivní „zmražení“ uživatele, ať už kvůli údržbě nebo například k bezpečnostnímu auditu. Vypnutí a opětovné zapnutí uživatele se provede následovně:

```
stack@home:~ $ openstack user set USER_NAME --disable
stack@home:~ $ openstack user set USER_NAME --enable
```

Příkaz 23: Zapnutí a vypnutí uživatele

[41]

Příkaz pro modifikaci uživatele využívá stejné syntaxe příkazu jako pro zapnutí a vypnutí uživatele jenom se přidají další detaily:

```
stack@home:~ $ openstack user set USER_NAME --email new-user@example.com
User has been updated.
```

Příkaz 24: Modifikace údajů uživatele

[41]

Odstranění uživatele se provede stejným příkazem, jako odebrání projektu, avšak se zaměněným klíčovým slovem user.

```
stack@home:~$ openstack user delete USER_NAME
```

Příkaz 25: Odstranění uživatele

[41]

7.3.3 Správa rolí

Pro plnohodnotné ovládání projektů a k nim přiřazeným uživatelům slouží role. Ty umožňují přidat dodatečné informace v rámci projektu, ke kterému se vážou. Výpis rolí se provede příkazem:

```
stack@home:~$ openstack role list
+-----+-----+
| ID                | Name          |
+-----+-----+
| 0278c36c00a140ddb | anotherrole   |
| f2eefe08845c390   | member       |
| 1def1796fb6345e9a | 70c60f740c9c |
| 4e14b088ee3762718 | 839d550d6c2e |
| 40b2920e4c422db52 | 8cc61b8933e4 |
| 43aaafe0a07020243 | c1a1ebeb7b08 |
| 487e85d6353c01d93 | e3cc958f66bc |
| bd2               | 4e6cb767f9c9 |
|                   | 8f04132d     |
|                   | service      |
+-----+-----+
```

Příkaz 26: Výpis rolí

[41]

Vytvoření role probíhá stejně jako ostatní příkazy pro vytvoření projektu či uživatele. Jedinou změnou je klíčové slovo `role`.

```
stack@home:~$ openstack role create ROLE_NAME
```

Příkaz 27: Vytvoření nové role

[41]

Hlavním důvodem existence rolí je jejich možnost přiřadit je určitým uživatelům, v rámci jednotlivého projektu. Přidání i odebrání specifické role, ke specifickému uživateli, se provede následujícím příkazem:

```
stack@home:~$ openstack role add --user USER_NAME \  
--project PROJECT_NAME ROLE_NAME  
stack@home:~$ openstack role remove --user USER_NAME \  
--project PROJECT_NAME ROLE_NAME
```

Příkaz 28: Přiřazení a odebrání role od uživatele

[41]

Vytváření a správa velkého počtu rolí, s faktem toho, že role jsou vázány k projektům, může vést k tomu, že velmi rychle naroste počet spravovaných rolí. Pro zjednodušení správy rolí lze vyvářet speciální pravidla, nazývané jako implikované role. Tyto role lze také jinak nazvat zřetězené, protože přiřazení jedné takové role zapříčiní i automatické přiřazení ostatních, implikovaných, rolí. Pokud jsou takové role odebrány, výsledkem je i ztráta implikovaných rolí. Vytvoření a odebrání implikovaných rolí se provede následovně:

```
stack@home:~$ openstack implied role create admin --implied-role member  
+-----+-----+  
| Field      | Value                                     |  
+-----+-----+  
| implies    | 71ccc37d41c8491c975ae72676db687f |  
| prior_role | 29c09e68e6f741afa952a837e29c700b |  
+-----+-----+  
stack@home:~$ openstack implied role delete admin --implied-role member
```

Příkaz 29: Vytvoření a odebrání implikovaných rolí

[41]

8 HORIZON

Pro snazší správu prostředků, infrastruktury a uživatelů je mnohem snazší používat grafické uživatelské rozhraní místo příkazové řádky, která může dělat problém některým uživatelům.

8.1 Instalace

Stejně jako modul Keystone lze nainstalovat modul nástěnky Horizon samostatně. V tomto případě to není zapotřebí, jelikož byl modul Horizon nainstalován za pomoci instalačního skriptu DevStack.

Pro manuální instalaci je nicméně zapotřebí mít certifikovaný systém, a to Debian, Ubuntu, (open)SUSE nebo RHEL, kde se veškeré balíky s jednotlivými moduly platformy OpenStack nacházejí. Instalace se provede stažením balíku `openstack-dashboard`, který spolu s sebou stáhne i HTTP server Apache2, na kterém daná nástěnka bude dostupná.

[42]

8.2 Spuštění

O spuštění nástěnky modulu Horizon se stará inicializační systém příslušného operačního systému Linux. Dnes nejčastěji `systemd`, avšak ve starších distribucích se může stále vyskytovat starší spouštěcí systém `SysVinit`.

[36], [43]

8.2.1 Potřebná konfigurace

Po manuální nebo balíkové instalaci je zapotřebí nakonfigurovat hned několik parametrů, a to konkrétně v souboru `/etc/openstack-dashboard/local_settings.py`. Pro správné fungování nástěnky je potřeba nakonfigurovat tyto parametry:

```
OPENSTACK_HOST = "controller"
ALLOWED_HOSTS = ['one.example.com', 'two.example.com']
SESSION_ENGINE = 'django.contrib.sessions.backends.cache'
CACHES = {
    'default': {
        'BACKEND': 'django.core.cache.backends.memcached.MemcachedCache',
        'LOCATION': 'controller:11211',
    }
}
OPENSTACK_KEYSTONE_URL = "http://%s/identity/v3" % OPENSTACK_HOST
OPENSTACK_KEYSTONE_MULTIDOMAIN_SUPPORT = True
OPENSTACK_API_VERSIONS = {
    "identity": 3,
    "image": 2,
    "volume": 3,
}
OPENSTACK_KEYSTONE_DEFAULT_DOMAIN = "Default"
OPENSTACK_KEYSTONE_DEFAULT_ROLE = "user"
TIME_ZONE = "TIME_ZONE"
```

Kód 7: Konfigurační proměnné nástěnky Horizon

[42]

Mějme na paměti, že toto nejsou proměnné prostředí, nýbrž se jedná o konfigurační proměnné, které jsou vloženy do konfiguračního skriptu jazyka Python 3, který umožňuje použít vnořené objekty a pole.

8.2.2 Parametry konfigurace

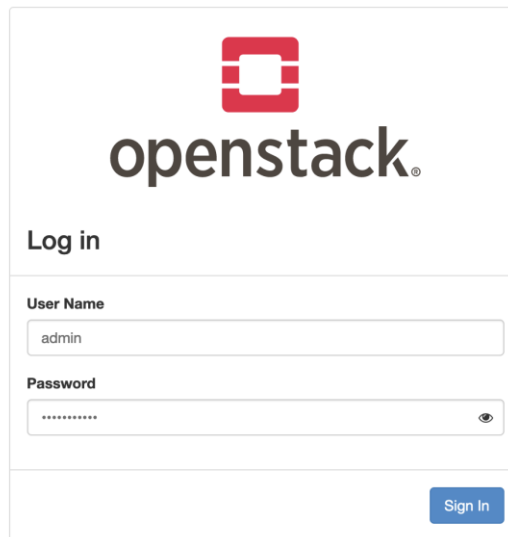
Tyto parametry je nutné nastavit podle výstupních dat, která jsou vypsána instalacemi jednotlivých modulů. Parametr `ALLOWED_HOSTS` lze také nastavit na hodnotu `['*']`, která znamená že se lze připojit k nástěnce modulu Horizon odkudkoliv, avšak je doporučeno omezit možnosti připojení pouze na povolené adresy IP. Ostatní konfigurační parametry je nutné zadat podle potřeb a konfigurace ostatních modulů, například vyžadovaného modulu Keystone.

8.3 Sledování stavu

Ve výchozím nastavení se nástěnka modulu Horizon nachází na adrese specifikovaného síťového rozhraní, případně na adrese IP, která byla specifikována v konfiguračním souboru.

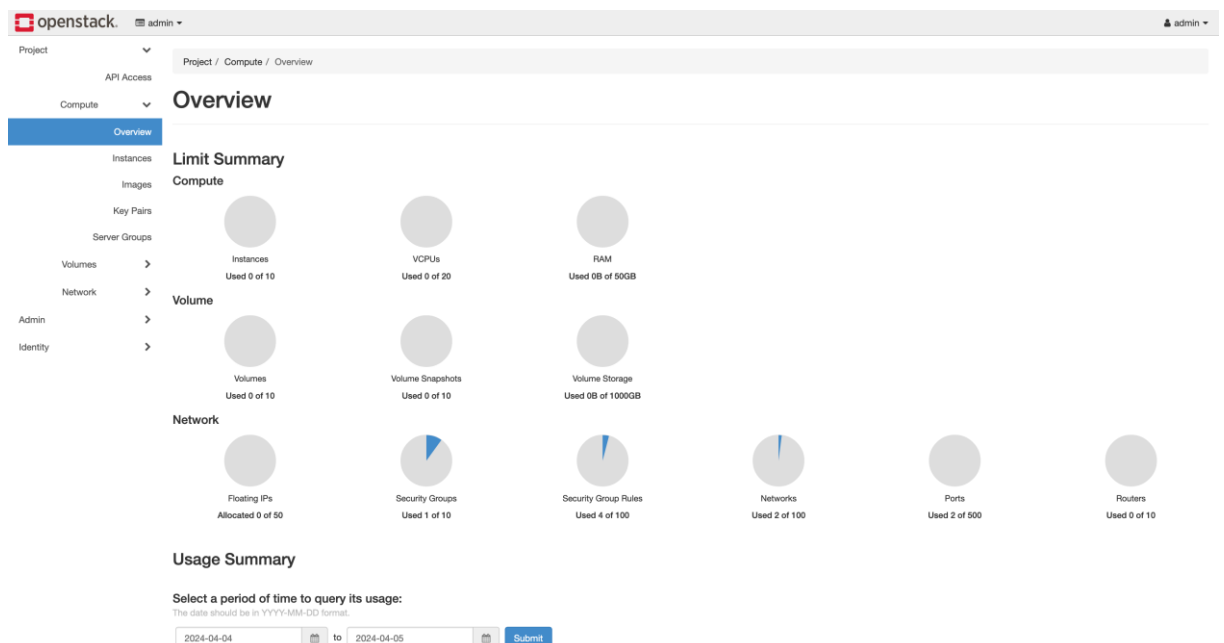
8.3.1 Přihlášení

Po zadání normálního připojení na adresu IP daného serveru, na kterém nástěnka Horizon běží, je uživateli předložena přihlašovací stránka, kam je uživatel požádán o přihlašovací jméno a heslo. Podle konfigurace v souboru `local.conf` (Kód 2).



Obrázek 2: Přihlašovací stránka nástěnky modulu Horizon

Po zadání přihlašovacího jména a hesla je uživatel přesměrován k přehledu nástěnky, kde je uveden současný stav platformy, s množstvím informací o platformě, běžících úkolech, hardwarových prostředcích a dalších administrativních informacích.



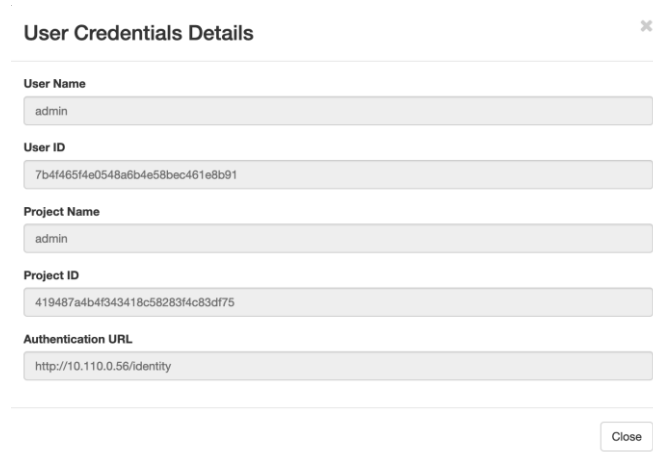
Obrázek 3: Přehled platformy OpenStack v nástěnce modulu Horizon

8.3.2 Přístup přes API

Pro přístup k platformě OpenStack lze použít i jednoduché požadavky protokolu HTTP. Takové požadavky typicky musí obsahovat jednu z podporovaných forem autentizace. Podporované autentizační formy jsou:

- přihlašovací jméno s heslem,
- přidělený token.

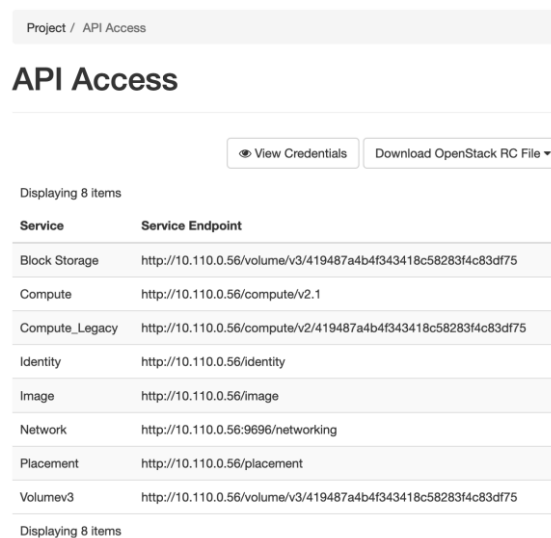
Z modulu Horizon lze vypsat přihlašovací údaje současně přihlášeného uživatele, v kategorii Project, podkategorie API access.



Field	Value
User Name	admin
User ID	7b4f465f4e0548a6b4e58bec461e8b91
Project Name	admin
Project ID	419487a4b4f343418c58283f4c83df75
Authentication URL	http://10.110.0.56/identity

Obrázek 4: Autentizační údaje, vypsané v nástěnce modulu Horizon

Na stejné stránce jsou uvedeny i spuštěné služby, na kterých naslouchají příslušné servery, příchozím požadavkům protokolu HTTP.



Service	Service Endpoint
Block Storage	http://10.110.0.56/volume/v3/419487a4b4f343418c58283f4c83df75
Compute	http://10.110.0.56/compute/v2.1
Compute_Legacy	http://10.110.0.56/compute/v2/419487a4b4f343418c58283f4c83df75
Identity	http://10.110.0.56/identity
Image	http://10.110.0.56/image
Network	http://10.110.0.56:9696/networking
Placement	http://10.110.0.56/placement
Volumev3	http://10.110.0.56/volume/v3/419487a4b4f343418c58283f4c83df75

Obrázek 5: Naslouchající služby protokolu HTTP.

8.4 Propojení s ostatními komponenty

Nástěnka modulu Horizon slouží jako centrální administrace pro celou platformu OpenStack, která je provozována na infrastruktuře. S vysokým počtem modulů je žádoucí umožnit připojit i jiné moduly do centralizované správy. Jak bylo popsáno v kapitole 8.3.2, přístup pro API je též centralizován a podle dostupných, nainstalovaných a podporovaných služeb.

Pro přístup k jednotlivým službám slouží naslouchající servery. Toto propojení umožňuje nejenom řídit, ale i sledovat průběh a stav vytížení infrastruktury, a to z hned mnoha pohledů, kde jsou sledovány relevantní údaje. Například pro modul Nova, výpočetní modul pro provoz virtuálních serverů, má k dispozici vlastní formu nástěnky, které běží pod modulem Horizon.

9 NOVA

Modul Nova, jenž se stará o provoz, instancování a mazání jednotlivých virtuálních serverů, které jsou jednou ze základních pracovních jednotek v platformě OpenStack.

9.1 Instalace

Samotná instalace modulu Nova je zpravidla vzácná, až na velmi specializované výpočetní servery (označovány jako compute node). Nicméně lze nainstalovat modul Nova ve dvou režimech, prvním je tzv. controller, ovladač, který zastupuje řídicí prvek v uzlech modulu Nova a v druhém režimu pracuje pouze jako „klient“ v infrastruktuře modulu Nova. Pro instalaci ovládacího modulu, s prvky pro ovládání celé infrastruktury modulu Nova, slouží softwarový balík nova, zatímco pro instalaci modulu Nova na uzel, který má být v rámci infrastruktury pouze v režimu výpočetního uzle, slouží instalační balík nova-compute.

Pro instalaci je nutné provést několik kroků, před samotnou instalací. Mezi ně patří:

- vytvoření a konfigurace databázi MySQL,
- vytvoření uživatele a přiřazení role admin uživateli nova,
- vytvoření naslouchajících serverů,
- instalace balíků modulu Nova,
- naplnění konfiguračních souborů,
- verifikace konfigurace,
- a finální restart služeb modulu Nova.

Vzhledem k faktu, že byl modul Nova nainstalován pomocí instalačních skriptů DevStack, není potřebná manuální instalace na jeden kontrolní server a je snazší a časově mnohem výhodnější, provést instalaci modulu Nova pomocí instalačního skriptu DevStack. Avšak pokud není žádoucí doinstalovat i ostatní komponenty, například na výpočetní uzel, je nutné nainstalovat modul Nova pomocí manuální instalace.

[44], [45]

9.2 Spuštění

Stejně jako u ostatních modulů, záleží na operačním systému, na kterém je služba spouštěna. Typicky se jedná o software systemd, který spouští služby, pokud jsou k tomu nakonfigurovány. Ve výchozím nastavení služeb jsou ihned po startu systému spuštěny.

9.2.1 Potřebná konfigurace

Pro funkčnost modulu Nova je zapotřebí jedna ze dvou konfigurací. Může se jednat o konfiguraci ovládacího uzlu nebo výpočetního uzlu. Pokud byl nainstalován modul Nova pomocí instalačního skriptu DevStack tak se jedná o ovládací uzel a konfigurace byla vytvořena automaticky. Nicméně pro konfiguraci manuálně instalovaného ovládacího uzlu je zapotřebí přidat vlastní konfigurační parametry do příslušných sekcí v souboru `/etc/nova/nova.conf`:

```
[api_database]
# ...
connection = mysql+pymysql://nova:NOVA_DBPASS@controller/nova_api

[database]
# ...
connection = mysql+pymysql://nova:NOVA_DBPASS@controller/nova

[DEFAULT]
# ...
transport_url = rabbit://openstack:RABBIT_PASS@controller:5672/
my_ip = 10.0.110.56

[api]
# ...
auth_strategy = keystone

[vnc]
enabled = true
# ...
server_listen = $my_ip
server_proxyclient_address = $my_ip

[oslo_concurrency]
# ...
lock_path = /var/lib/nova/tmp
```

Kód 8: Konfigurace vlastních parametrů ovládacího uzlu modulu Nova

Pro konfiguraci ovládacího uzlu je zapotřebí také konfigurace dodatečných parametrů u modulů, na kterých je modul Nova závislý. Pro nastavení autentizace, přes modul Keystone, je zapotřebí přidat následující parametry:

```
[api]
# ...
auth_strategy = keystone

[keystone_authtoken]
# ...
www_authenticate_uri = http://controller:5000/
auth_url = http://controller:5000/
memcached_servers = controller:11211
auth_type = password
project_domain_name = Default
user_domain_name = Default
project_name = service
username = nova
password = NOVA_PASS

[service_user]
send_service_user_token = true
auth_url = https://controller/identity
auth_strategy = keystone
auth_type = password
project_domain_name = Default
project_name = service
user_domain_name = Default
username = nova
password = NOVA_PASS
```

Kód 9: Konfigurace parametrů Keystone u ovládacího uzlu modulu Nova

[44]

Dodatečně je zapotřebí přidat konfiguraci úložných modulu Glance a modulu Placement, který není nainstalován pomocí skriptu DevStack.

```
[glance]
# ...
api_servers = http://controller:9292

[placement]
# ...
region_name = RegionOne
project_domain_name = Default
project_name = service
auth_type = password
user_domain_name = Default
auth_url = http://controller:5000/v3
username = placement
password = PLACEMENT_PASS
```

Kód 10: Konfigurace parametrů Glance a Placement u ovládacího uzlu modulu Nova

[44]

9.3 Nástěnka modulu Horizon

Modul Nova má k dispozici ovládací prostředí v nástěnce modulu Horizon. V tomto rozhraní lze vytvářet nové instance virtuálních serverů, sledovat jejich provoz, upravovat je během běhu nebo nečinnosti a jejich následné vymazání.

9.3.1 Administrativní nástěnka

Pro administraci, kde se uživatel musí přihlásit jako administrátor (uživatel s rolí admin), k přístupu k nástěnce administrátora slouží záložka modulu Nova, pojmenovaná jako Compute. Tato záložka má další podzáložky:

- Hypervisors,
- Host Aggregates,
- Instances,
- Flavors,
- Images.

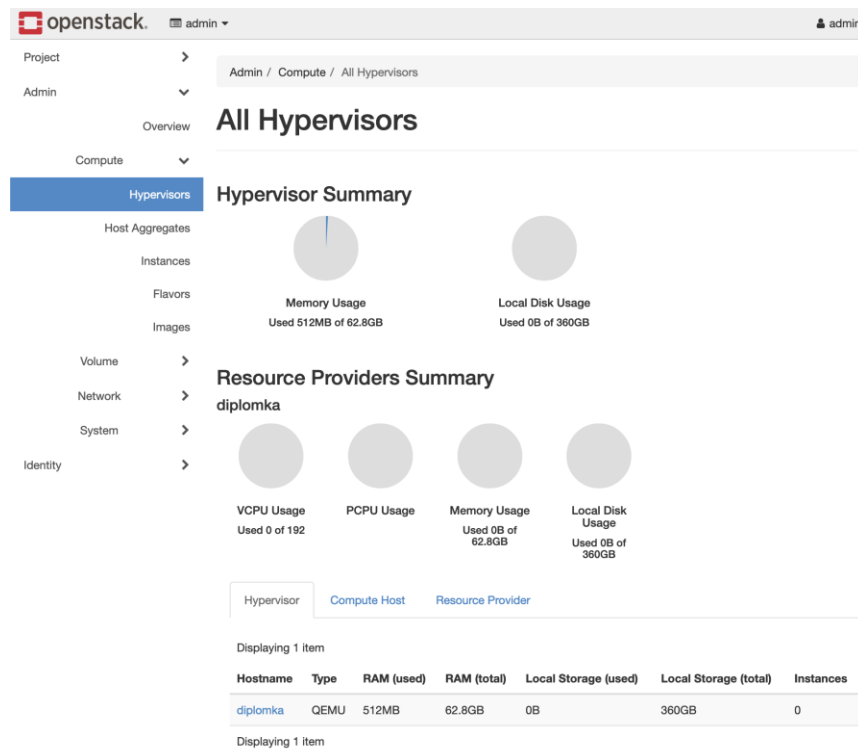
Tyto záložky slouží k plnohodnotné administraci modulu Nova, pomocí uživatelského rozhraní nástěnky modulu Horizon.

9.3.2 Hypervizory

Nástěnka s hypervizory vypisuje informace o celkovém využití prostředků, běžících instancí, připojených uzlů společně s typem virtualizace a dostupném úložném místě. Sjednocuje pod sebe celý výpočetní cluster, z celé infrastruktury a vypisuje veškerá potřebná data pro snadné monitorování.

V další kartě na nástěnce, pojmenované jako Compute Host jsou vypsány všechny uzly, které poskytují svůj hardware pomocí modulu Nova, ať už samotný kontrolní uzel, tak i ostatní, čistě výpočetní uzly, s nainstalovaným a nakonfigurovaným balíkem nova-compute.

V třetí a poslední kartě, pojmenované Resource Provider jsou vypsány jednotlivé hardwarové prostředky, které jsou z připojených uzlů poskytovány. Vypsány jsou přímo podle uzlů, na kterých se daná zařízení a hardware nachází. Tento hardware, pokud je rozdělen na virtuální a fyzický, je označován s předponou písmene P (označující fyzický hardware, z anglického physical), kde například PCPU je počet opravdových jader (případně vláken u více vláknového procesoru), kterými daný uzel disponuje.



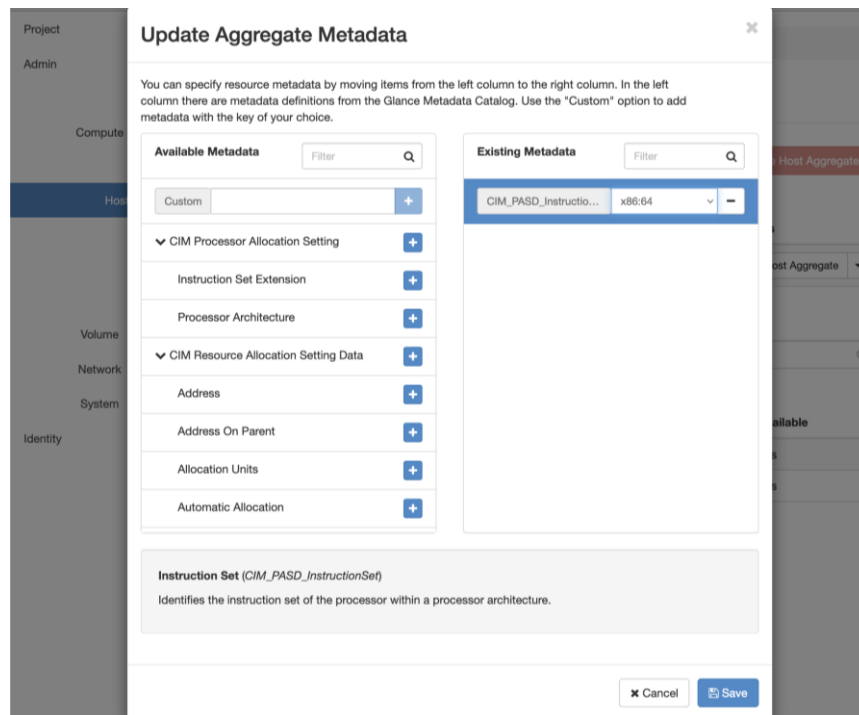
Obrázek 6: Nástěnka hypervisorů modulu Nova

9.3.3 Host Aggregates

Další nástěnkou modulu Nova je záložka Host Aggregates, jenž se stará o seskupování virtuálních serverů (pojmenovaných jako Host, ze stejného anglického slova) do sjednocených celků. Tyto celky mají za cíl sjednocovat uzly pro běh specifických virtuálních serverů, vyžadující specifické podmínky na hardware. Například lze vytvářet agregace uzlů, které mají úložiště na discích SSD, místo pomalých disků HDD.

Pro vytvoření agregace je zapotřebí pouze název agregace. Po vytvoření lze přejít k aktualizaci požadovaných metadat, kde lze určit jednotlivé požadavky a čím se vyznačují.

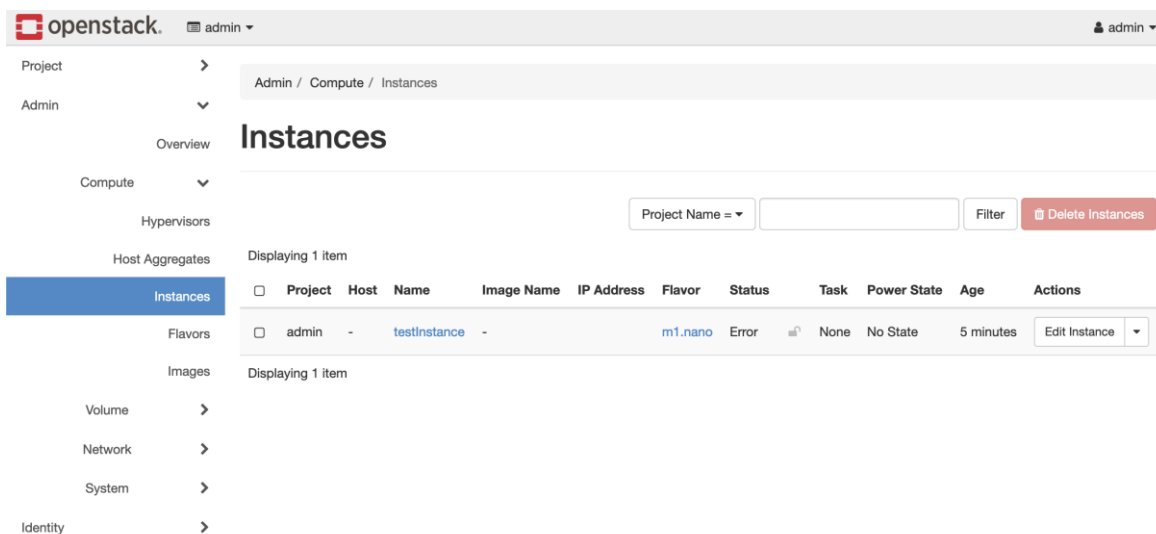
[46]



Obrázek 7: Metadata agregací jednotlivých uzlů a jejich filtrování

9.3.4 Instances

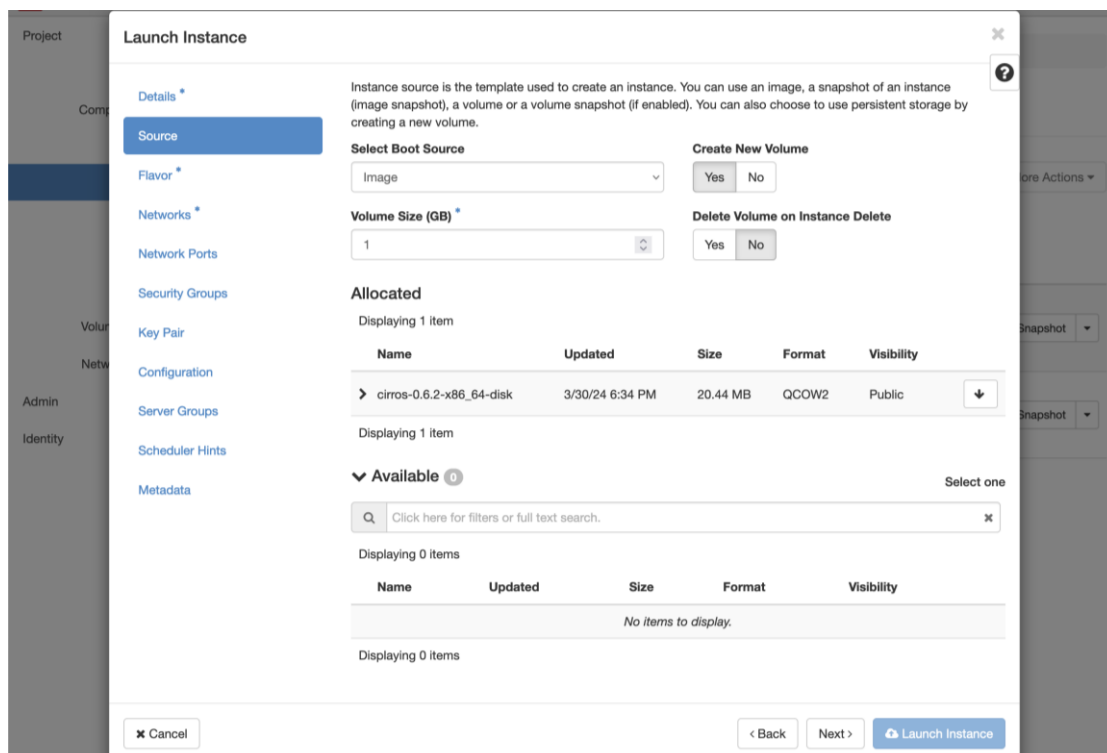
Nástěnka Instances vypisuje vytvořené instance, jejich přiřazení k projektům, jaký obraz je v instanci spuštěn a další informace, které jsou důležité pro sledování a provoz virtuálních serverů. Vytvořené instance nelze po vytvoření upravovat, a tak je nutné při jejich vytváření dbát na požadovanou konfiguraci.



Obrázek 8: Vytvořené instance modulu Nova

Při vytváření instancí je nutné specifikovat parametry, jaké budou použity k vytvoření instance. Mezi tyto parametry patří:

- jméno instance a jejich počet,
- zdrojový obraz instancí a přiřazenou velikost úložišť,
- flavor, podle kterého jsou přiřazeny prostředky jako jsou CPU a RAM,
- síť do které bude instance připojena.



Obrázek 9: Vytváření instancí virtuálních serverů v nástěnce modulu Nova

9.3.5 Flavors

Nástěnka Flavors se stará o vytváření a správu alokovatelných konfigurací. Typicky se jedná o maximální přiřaditelnou velikost paměti RAM, počet GB výchozího úložiště pro daný obraz, odkládacího úložiště (swap), počet virtuálních jader CPU či poměr rychlosti připojení. Platforma OpenStack přichází s přednastavenou množinou těchto konfigurací. Tyto jednotky konfigurací se přiřazují jednotlivým instancím, které jsou vytvářeny.

9.4 Problémy s modulem Nova

Po instalaci modulu Nova a prvotním spuštěním dvou testovacích instancí přibaleného obrazu operačního systému CirrOS vzniklo několik problémů.

9.4.1 Nemožnost spuštění jakékoliv instance

Po instalaci platformy OpenStack a jejím spuštění vyšlo najevo že nelze spustit jakoukoliv instanci modulu Nova, která by běžela na svém vlastním diskovém prostoru, typicky uchovávaného v modulu Cinder. Řešením tohoto problému je konfigurace dodatečného úložiště, konkrétněji popsána v kapitole 6.2.4, kde je konkrétně popsán problém s úložištěm a způsobem, jak takový problém řešit.

9.4.2 Nemožnost startu instancí po restartu systému

Po restartu systému, ať už plánovaného nebo neplánovaného, se odmítne spustit služba pro správu CPU zdrojů modulu Nova, konkrétně pojmenovaná jako `devstack@n-cpu.service`. Tato služba je nedílnou součástí virtualizační platformy modulu Nova a pro fungování jakékoliv instance musí tato služba běžet. Problémem je velmi brzký start služby, který je proveden ještě dříve, než se stihne modul Keystone, na kterém je modul Nova závislý, plně spustit. Jedním řešením je přidat následující kód, který zpomalí spuštění dané služby o 5 sekund pomocí příkazu `sleep`, do souboru `devstack@n-cpu.service`:

```
[Service]
# ...
ExecStartPre=/usr/bin/sleep 5
```

Kód 11: Konfigurace parametrů Glance a Placement u ovládacího uzlu modulu Nova

Po restartu systému se daná služba spustí správně. Dalším způsobem, který by mohl fungovat, by bylo zavést parametr `After`, který by zajistil že by se daná služba spustila až teprve po dané službě, v tomto případě službě modulu Keystone. Po experimentování bylo, ale v některých případech zjištěno, že i po opožděném spuštění za pomocí parametru `After`, nebyla služba modulu Nova korektně spuštěna, pravděpodobně kvůli problému souběhu, kde se služba modulu Nova začne spouštět sice po startu modulu Keystone, avšak modul Keystone ještě není plně nainicializován na to, aby služba modulu Nova byla správně připojena k modulu Keystone.

```
LOAD ACTIVE SUB DESCRIPTION
loaded active running Devstack devstack@n-api.service
loaded active running Devstack devstack@n-sch.service
loaded active running Devstack devstack@n-vol.service
loaded active running Devstack devstack@n-dstat.service
loaded active running Devstack devstack@n-etc.service
loaded active running Devstack devstack@n-g-api.service
loaded active running Devstack devstack@n-keystone.service
loaded active running Devstack devstack@n-api-meta.service
loaded active running Devstack devstack@n-api.service
loaded active running Devstack devstack@n-cond-cell11.service
loaded failed failed Devstack devstack@n-cpu.service
```

Obrázek 10: Selhání služby `n-cpu` modulu Nova po startu systému

9.4.3 Nemožnost pokračování instance z jejího pozastavení

Po pozastavení jedné z testovacích instancí (konkrétně příkaz Suspend), nebylo možné dále spustit (příkazem Resume). Řešení přes nástěnku modulu Horizon nebylo možné, z důvodu nemožnosti provést příkaz Hard Reboot. Ten lze provést přes příkazovou řádku za pomoci následujícího příkazu, ke místo parametru SERVER se uvede název instance:

```
stack@home:~$ openstack server reboot --hard SERVER
```

Příkaz 30: Vynucený restart instance, která se není schopna spustit příkazem Resume

10 GLANCE

Modul Glance se stará o správu obrazů, jejich vytváření, metadata a mazání, kde umožňuje v rámci infrastruktury OpenStack jejich instancování modulem Nova.

10.1 Instalace

Stejně jako ostatní moduly, které byly nainstalovány instalačním skriptem DevStack, proběhla instalace automatizovaně. Nicméně lze tento modul nainstalovat ručně, pomocí balíčkovacího systému dané distribuce operačního systému Linux. Pro instalaci je tedy zapotřebí operační systém Debian, Ubuntu, RHEL či CentOS. Samotná instalace proběhne pomocí nainstalování balíčku glance. Předem je, ale nutné vytvořit účty pro připojení modulu Glance do infrastruktury OpenStack:

```
stack@home:~$ openstack user create --domain default --password-prompt glance
... výstup příkazu

stack@home:~$ openstack role add --project service --user glance admin
... výstup příkazu

stack@home:~$ openstack endpoint create --region RegionOne \
    image public http://controller:9292
... výstup příkazu

stack@home:~$ openstack endpoint create --region RegionOne \
    image internal http://controller:9292
... výstup příkazu

stack@home:~$ openstack endpoint create --region RegionOne \
    image admin http://controller:9292
... výstup příkazu

lukas@home:~# apt install glance
```

Příkaz 31: Vytvoření a přidání uživatele glance a jeho rolí, pro instalaci modulu Glance

[47]

10.2 Spuštění

Jako u ostatních modulů se o start stará automatizovaný systém. Dle distribucí se jedná buďto o systém systemd nebo SysVinit. Spouštění služeb je prováděno po inicializaci systému, ihned jak je to možné a s co nejnižší prodlevou.

[36], [41]

10.2.1 Potřebná konfigurace

Pro spuštění a běh modulu Glance je nutné přidat do konfigurace několik parametrů, podobně jako je tomu u modulu Nova. Tyto parametry není nutné přidávat, pokud byl modul Glance nainstalován pomocí instalačního skriptu DevStack. Přidané parametry se vkládají do souboru `/etc/glance/glance-api.conf`:

```
[database]
connection = mysql+pymysql://glance:GLANCE_DBPASS@controller/glance

[paste_deploy]
flavor = keystone

[DEFAULT]
enabled_backends=fs:file

[glance_store]
# ...
default_backend = fs

[fs]
filesystem_store_datadir = /var/lib/glance/images/
```

Kód 12: Vkládané parametry pro konfiguraci modulu Glance

[47]

Pro správné fungování modulu Glance je nutné přidat i konfigurační sekci pro modul Keystone:

```
[keystone_authtoken]
# ...
www_authenticate_uri = http://controller:5000
auth_url = http://controller:5000
memcached_servers = controller:11211
auth_type = password
project_domain_name = Default
user_domain_name = Default
project_name = service
username = glance
password = GLANCE_PASS

[oslo_limit]
auth_url = http://controller:5000
auth_type = password
user_domain_id = default
username = glance
system_scope = all
password = GLANCE_PASS
endpoint_id = 340be3625e9b4239a6415d034e98aace
region_name = RegionOne
```

Kód 13: Konfigurace parametrů Keystone u ovládacího uzlu modulu Glance

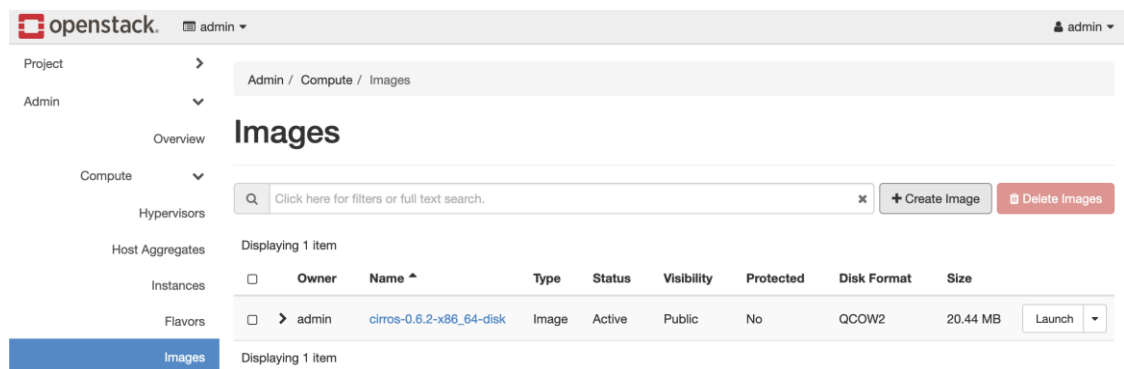
[47]

10.3 Práce s obrazy virtuálních strojů

Práce s obrazy probíhá nejčastěji přes nástěnku, dostupnou přes modul Horizon. Ta usnadňuje práci s obrazy, které lze následně v modulu Nova instancovat.

10.3.1 Přidání

Pro přidání obrazu je zjednodušeno pomocí nástěnky a webového rozhraní. Záložka Images, jenž se nachází pod kartou Compute modulu Nova, se stará o správu obrazů. Přestože je tato záložka pod modulem Nova, tak se jedná o přímou správu modulu Glance, na kterém je modul Nova závislý. Správa obrazů se provádí typicky pomocí nástěnky v modulu Horizon.



Obrázek 11: Správa obrazů pro virtuální servery, pocházející z modulu Cinder

Vytvářet lze nové obrazy tak, že se před nasazením vytvoří příslušný obraz nainstalovaného systému, tak jak by byl spuštěn v klasickém virtuálním prostředí jiných virtuálních služeb či programů. Například lze vytvořit obraz virtualizační platformy QEMU, který ve výchozím nastavení používá diskový formát QCOW2. Po nastavení a konfiguraci systému lze tento systém zabalit nebo čistě poslat do modulu Glance, kde lze tento obraz nakopírovat a spustit v samostatných instancích virtuálních serverů v rámci modulu Nova. Pro nahrání nového obrazu je možné použít několik formátů, z nichž mezi neznámější se řadí:

- ISO,
- QCOW2,
- VMDK,
- VHD.

Obrázek 12: Vytvoření a nahrání nového obrazu do modulu Glance

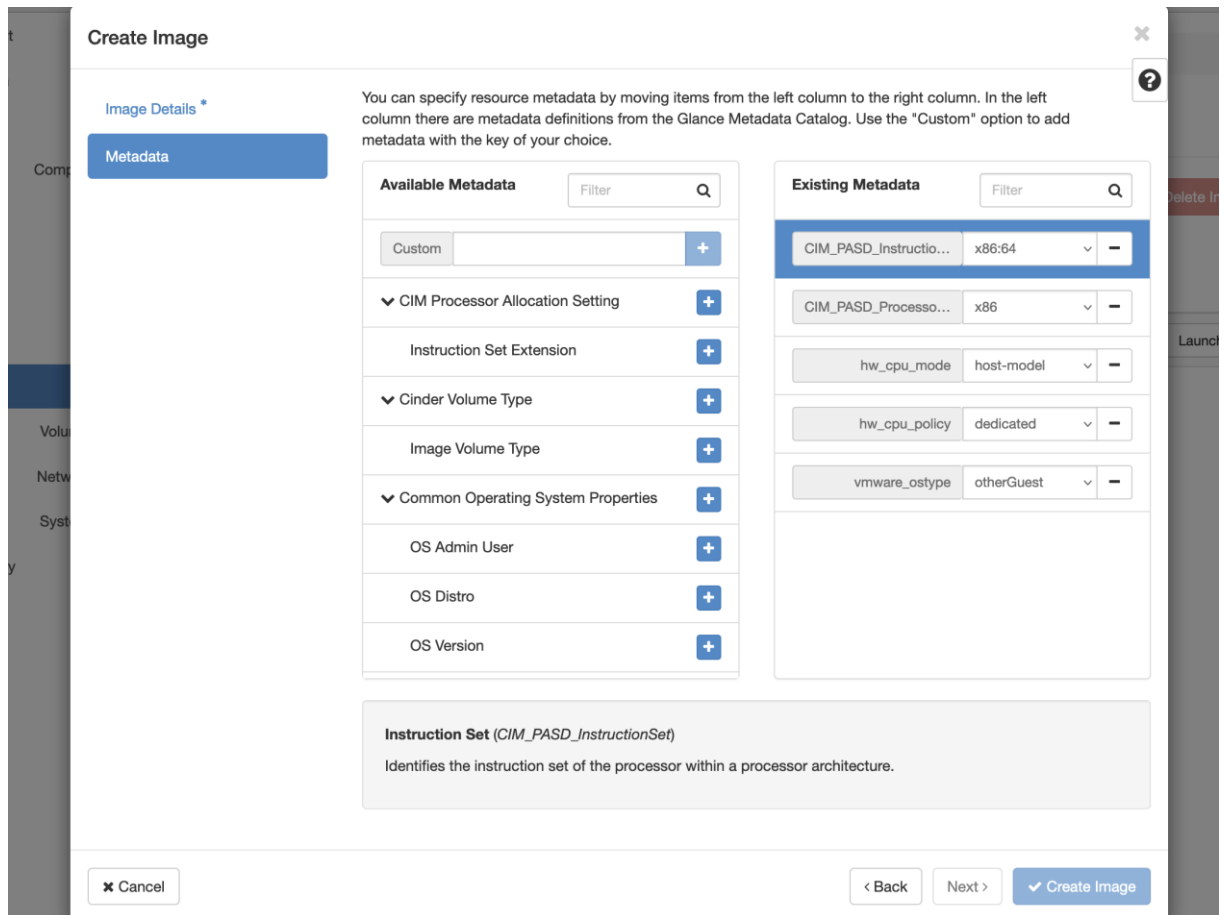
Při vytváření obrazu lze specifikovat dodatečné parametry, ať už v rámci přímého popisu obrazu, jako jsou parametry kernel, ramdisk, architektura, minimální velikost kořenového disku či minimální požadovaná paměť RAM.

10.3.2 Správa metadat

Modul Glance umožňuje uvádět dodatečná metadata u obrazů. V základním nastavení je uvedeno několik kategorií metadat, jako jsou například:

- architektura CPU,
- požadovaný instrukční set,
- přiřazování CPU jader,
- databázový software,
- maximální limity VCPU,
- distribuce operačního systému,
- verze operačního systému.

Správa metadat probíhá druhé záložky při vytváření obrazu nebo při jeho editaci v záložce Images. Všechna tato metadata jsou buďto výčtového typu, kde specifikujeme jednu hodnotu z množiny povolených hodnot nebo doplňková, kde doplňujeme text či čísla, která odpovídají významu metadatům.



Obrázek 13: Správa metadat v nástěnce modulu Glance

10.3.3 Odebrání obrazů

Odebrání obrazu je destruktivní operace, u které odebereme daný zdrojový obraz, čímž se stane nadále nedostupným. Vytvořené instance tohoto obrazu touto operací nebudou nijak ovlivněny, protože vytvořená instance má vlastní vyhrazený úložný prostor, typicky v modulu Cinder, ve kterém se nachází kopie celého obrazu po jeho instancování. Avšak změny v určité instanci jednoho z obrazů mohou vést k velmi odlišné konfiguraci a „znečištění“ vlastního obrazu natolik, že jeho použití pro vytvoření nového obrazu není dostačující.

11 SWIFT

Modul Swift se stará o objektové úložiště. Takové úložiště slouží k ukládání rozmanitých dat pod jednotlivé účty a kontejnery, v hierarchické úrovni.

11.1 Instalace

Instalace modulu Swift není ve výchozí konfiguraci zajištěna pomocí instalačního skriptu DevStack, nicméně lze modul Swift pomocí instalačního skriptu DevStack doinstalovat, a to i po prvotní instalaci. Prvním krokem k automatizované instalaci je přidáním následující konfigurace do souboru `/opt/stack/devstack/local.conf`:

```
[[local|localrc]]
# ...
enable_service s-proxy s-object s-container s-account
SWIFT_HASH=UPCEst58238
```

Kód 14: Přidání požadavku na instalaci modulu Swift v souboru `local.conf`

[48]

Po opětovaném spuštění instalačního skriptu DevStack se provedou dodatečné změny. Provedou se tedy aktualizace a dodatečná instalace nebo odinstalace ostatních modulů či služeb, zde konkrétně specifikovány všechny služby kontrolního uzlu modulu Swift.

```
+functions-common:git_clone:600 RECLONE=False
+functions-common:git_clone:601 [[ 0 -gt 0 ]]
+functions-common:git_clone:605 [[ False = \T\r\u\e ]]
+functions-common:git_clone:614 echo master
+functions-common:git_clone:614 egrep -q '^refs'
+functions-common:git_clone:628 [[ ! -d /opt/stack/swift ]]
+functions-common:git_clone:629 [[ False = \T\r\u\e ]]
+functions-common:git_clone:634 git_timed clone --no-checkout https://opendev.org/openstack/swift.git /opt/stack/swift
+functions-common:git_timed:701 local count=0
+functions-common:git_timed:702 local timeout=0
+functions-common:git_timed:704 [[ -n 0 ]]
+functions-common:git_timed:705 timeout=0
+functions-common:git_timed:708 time_start git_timed
+functions-common:time_start:2407 local name=git_timed
+functions-common:time_start:2408 local start_time=
+functions-common:time_start:2409 [[ -n '' ]]
+functions-common:time_start:2412 date +%s%3N
+functions-common:time_start:2412 _TIME_START[$name]=1714150566972
+functions-common:git_timed:709 timeout -s SIGINT 0 git clone --no-checkout https://opendev.org/openstack/swift.git /opt/stack/swift
Cloning into '/opt/stack/swift'...
remote: Enumerating objects: 101800, done.
remote: Counting objects: 100% (63098/63098), done.
remote: Compressing objects: 100% (22436/22436), done.
```

Obrázek 14: Probíhající klonování repozitáře modulu Swift

Pokud instalace neproběhne v pořádku, a výstupním kódem není nula, tak instalace v některé části selhala. Pokud k tomuto dojde tak je vysoce doporučeno spustit odinstalaci pomocí instalačního skriptu DevStack, a to konkrétně následujícími příkazy:

```
stack@home:~/devstack$ ./unstack.sh; ./clean.sh; ./stack.sh
```

Příkaz 32: Přeinstalace softwaru OpenStack

11.2 Spuštění

O spuštění se stejně jako u ostatních modulů stará inicializační systémový program. Typicky buďto `systemd` nebo `SysVInit`.

[36], [43]

11.2.1 Parametry runtime konfigurace

Veškerá potřebná konfigurace byla již zadána před instalací. Nicméně, pokud nebude zadán parametr `SWIFT_HASH` tak bude instalační skript pozastaven, aby byl tento údaj doplněn. Primární konfigurační parametry, potřebné pro provoz modulu Swift se nachází v souboru `/etc/swift/swift.conf`:

```
[swift-constraints]
max_file_size = 5368709122      # Maximální velikost objektu, v bajtech
max_header_size = 16384        # Maximální bajtový objem hlavičky objektu

[swift-hash]
swift_hash_path_suffix = UPCEst58238 # Zadaný hash prefix pro instalaci
swift_hash_path_prefix = changeme    # Změnit na hash suffix

[storage-policy:0]
name = Policy-0                # Pojmenování úkládací politiky
default = yes                  # Jedná se o výchozí politiku
aliases = yellow, orange      # Zástupné aliasy politiky
```

Kód 15: Ukázka parametrů konfigurace objektového úložiště modulu Swift

11.3 Datové prostory

Pro ukládání jednotlivých objektů slouží jednotlivé účty, ke kterým jsou přiřazené jednotlivé kontejnery, v nichž se nacházejí uložené objekty. Správa datových prostorů nejčastěji probíhá přes nástěnku modulu Horizon.

11.3.1 Vytvoření

Vzhledem k faktu, že se objektové úložiště řeší na úrovni projektů, tak není dostupná vlastní záložka mimo projektovou kartu. V první úrovni se musí vytvořit kontejner, který je poté přístupný v rámci účtu, pokud je privátní, nebo pro kohokoliv v projektu, pomocí sdílením přístupového odkazu. Sdílení lze měnit i po vytvoření.

Pro vytvoření kontejneru je zapotřebí mít aktivní projekt, ve kterém daný účet figuruje. Kontejnery jsou vždy přiřazeny k účtům, ale pouze v rámci projektu.

Create Container ✕ ?

Container Name *

testContainer-2 ✓

Container name must not contain "/".

Storage Policy *

Policy-0 ▾

Container Access

Public Not public

A Public Container will allow anyone with the Public URL to gain access to your objects in the container.

✕ Cancel ✓ Submit

Obrázek 15: Vytvoření nového kontejneru modulu Swift ve vybraném projektu

Po vytvoření lze spravovat určený kontejner, stejně jako procházet jeho obsah. Kontejner je ve své podstatě organizované (uživatelské) úložiště se strukturou adresářů, jako jsou klasické souborové systémy. V názvu kontejneru mohou být veškeré znaky, vyjma znaku lomítka, které slouží k oddělování adresářů, podadresářů a souborů.

11.3.2 Správa

Pro správu slouží vlastní a jediná karta modulu Swift. Ta umožňuje procházet jednotlivé kontejnery, nacházející se pod přihlášeným účtem a specifickým projektem. Zároveň je u výpisu stavu kontejneru měnit jeho viditelnost (soukromý nebo veřejný/sdílený).

Project / Object Store / Containers

Containers

+ Container

Q Click here for filters or ft. ✕

testContainer-1

Object Count: 1
Size: 582.67 KB
Date Created: Apr 26, 2024
Storage Policy: Policy-0
Public Access: [Link](#)

testContainer-1

Q Click here for filters or full text search. ✕ 📁 + Folder 🗑️

Displaying 1 item

<input type="checkbox"/>	Name ^	Size
<input type="checkbox"/>	testFolder	Folder 🗑️ Delete

Displaying 1 item

Obrázek 16: Správa kontejnerů ve specifickém projektu

11.3.3 Nahrávání souborů

Po otevření vybraného kontejneru lze nahrát potažmo jakýkoliv soubor. Existují určitá omezení, která jsou nastavena pomocí konfiguračního souboru, například maximální velikost ukládaného objektu.

Pro hierarchickou strukturu objektového úložiště lze vytvářet adresáře. Nástěnka modulu Horizon, která pracuje nad modulem Swift má k dispozici tlačítko pro vytvoření adresářů i bez nahrávání souborů. Avšak při nahrávání souboru lze také vytvářet kompletní hierarchickou cestu k souboru, a to včetně vnořených adresářů, bez nutnosti jejich manuálního vytváření. Pro vytvoření hierarchické cesty stačí při nahrávání souboru vložit před název souboru názvy adresářů, oddělených lomítky.

Upload File To: testContainer-1 ✕

File *

Analýza.docx

File Name

Note: Delimiters ('/') are allowed in the file name to place the new file into a folder that will be created when the file is uploaded (to any depth of folders).

Obrázek 17: Nahrání nového souboru společně s jeho hierarchickou cestou

12 NEUTRON

Modul neutron se stará o vytváření, nastavení a kompletní správu síťových komponent ve stylu softwarově definovaných sítí (SDN).

12.1 Instalace

Modul neutron je nainstalován pomocí instalačního skriptu DevStack. Pro nasazované prostředí není nutná další manuální konfigurace a modul Neutron je nainstalován v režimu ovládacího uzlu (v anglické literatuře controller node). Nicméně pro manuální instalaci je nutné provést několik kroků, kde prvním z nich je vytvoření databáze neutron v databázovém softwaru MySQL nebo MariaDB. Poté je nutné provést několik příkazů ke konfiguraci modulu Neutron, v platformě OpenStack.

```
stack@home:~$ openstack user create --domain default --password-prompt neutron
... výstup příkazu

stack@home:~$ openstack role add --project service --user neutron admin
... výstup příkazu

stack@home:~$ openstack service create --name neutron \
--description "OpenStack Networking" network
... výstup příkazu

stack@home:~$ openstack endpoint create --region RegionOne \
network public http://controller:9696
... výstup příkazu

stack@home:~$ openstack endpoint create --region RegionOne \
network internal http://controller:9696
... výstup příkazu

stack@home:~# openstack endpoint create --region RegionOne \
network admin http://controller:9696
```

Příkaz 33: Vytvoření a přidání uživatele neutron a sítí, pro instalaci modulu Neutron

[49], [50]

12.2 Spuštění

Po instalaci a konfiguraci modulu Neutron je před spuštěním nutné přidat konfigurace do příslušných souborů.

12.2.1 Potřebná konfigurace

Jednu z potřebných konfigurací je nutné přidat do souboru `/etc/neutron/metadata_agent.ini`, kde místo parametru `METADATA_SECRET` vložíme jakékoliv heslo.

```
[DEFAULT]
# ...
nova_metadata_host = controller
metadata_proxy_shared_secret = METADATA_SECRET
```

Kód 16: Vkládané parametry pro konfiguraci metadat modulu Neutron

[49]

A do druhého konfiguračního souboru `/etc/nova/nova.conf`, tentokrát do konfiguračního souboru modulu Nova, je nutné přidat následující konfigurační parametry. Je nutno podotknout že je nutné doplnit vlastní údaje místo parametrů `NEUTRON_PASS` a `METADATA_SECRET`.

```
[neutron]
# ...
auth_url = http://controller:5000
auth_type = password
project_domain_name = Default
user_domain_name = Default
region_name = RegionOne
project_name = service
username = neutron
password = NEUTRON_PASS
service_metadata_proxy = true
metadata_proxy_shared_secret = METADATA_SECRET
```

Kód 17: Vkládané parametry modulu Neutron pro konfigurační soubor modulu Nova

[49]

12.2.2 Start služby neutron

Po nastavení konfiguračních souborů, vytvoření uživatele a sítě, společně s přiřazením rolí lze přejít ke spuštění. Finální nastavení modulu Neutron se provede pomocí následujícího příkazu:

```
stack@home:~$ su -s /bin/sh -c "neutron-db-manage \
--config-file /etc/neutron/neutron.conf \
--config-file /etc/neutron/plugins/ml2/ml2_conf.ini \
upgrade head" neutron
```

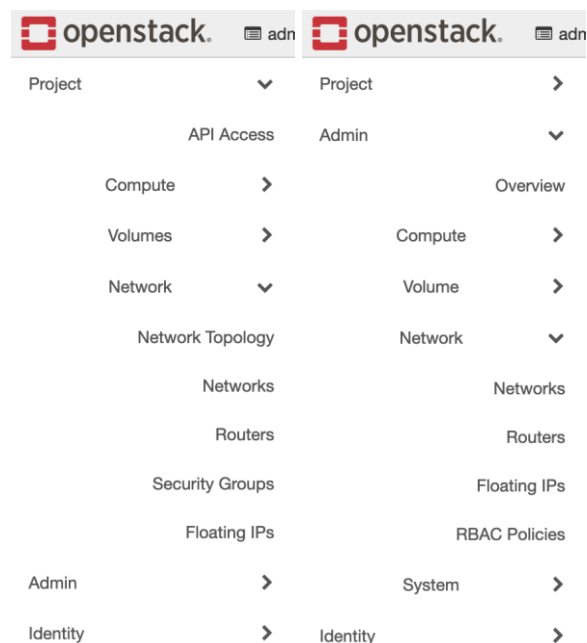
Příkaz 34: Finální příprava databáze modulu Neutron

12.3 Síť

Pro správu sítí modulu Neutron se typicky používá nástěnka modulu Horizon, kde jsou dané sítě vizualizovány, vytvářeny, modifikovány, přiřazovány a odebírány. Celý systém síťování napříč infrastruktur, pokud je zaštitěn modulem Neutron, probíhá automaticky, bez manuálního zásahu do konfiguračních souborů.

12.3.1 Nástěnka modulu Neutron

Pro potřeby nástěnky modulu Neutron slouží vlastní záložka pojmenovaná Network, a to jak v určitém projektu, tak i v administrátorské záložce.



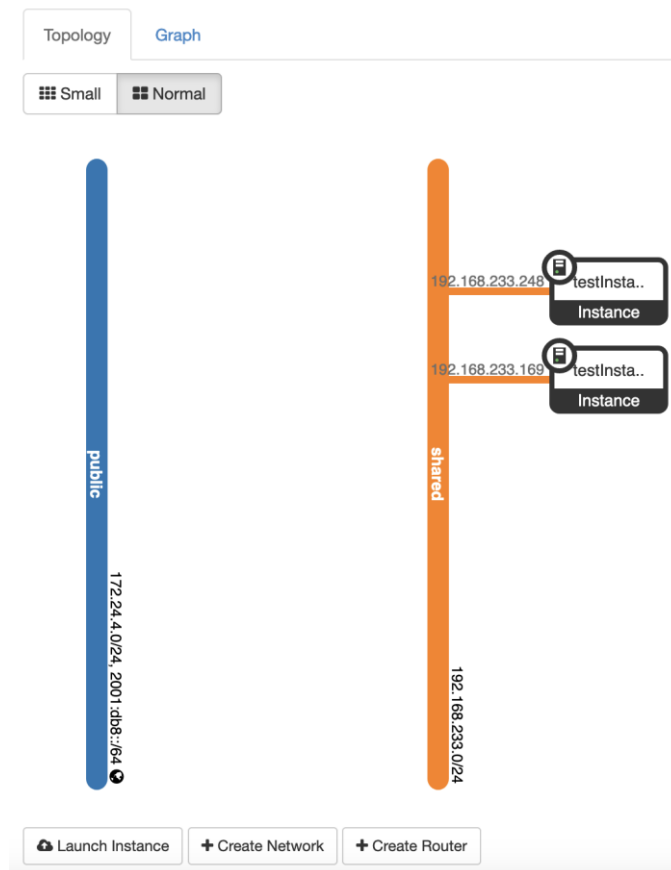
Obrázek 18: Nástěnky modulu Neutron v nástěnce Horizon

V nástěnce, v rámci specifického projektu lze také vizualizovat dostupné sítě a instance, které k ní jsou připojené. Jednotlivé záložky slouží ke správě či monitorování sítí v rámci modulu Neutron a platformy OpenStack.

12.3.2 Topologie sítě

V nástěnce lze vizualizovat topologii sítě. Nejedná se o klasické topologie typu hvězda, mesh, strom a další taxonomie sítí, nýbrž o připojení instancí k jednotlivým sítím, jejich asociovanou adresu IP a přehled stavu. Toto slouží k monitorování a zjišťování spojení jednotlivých instancí a sítí.

Network Topology



Obrázek 19: Přehled topologie sítí v nástěnce modulu Neutron

Dodatečně lze pro současně vybraný projekt vytvořit novou síť nebo směrovač, který se bude starat o propojení těchto sítí.

12.3.3 Vytvoření sítě

Pro vytvoření sítě je zapotřebí několik údajů, které jsou pro chod sítě nezbytné. Mezi takové údaje patří název sítě, zóny, do kterých bude síť spadat, nastavení, zdali je síť sdílená, parametr MTU. Dodatečně je možné manuálně vytvořit i podsítě, která bude síti náležet a lze u této podsítě manuálně přiřadit i adresu, protokol IP verze 4 nebo 6, a dodatečně i výchozí bránu.

Create Network ✕

Network
Subnet
Subnet Details

Network Name

Create a new network. In addition, a subnet associated with the network can be created in the following steps of this wizard.

Enable Admin State ⓘ

Shared

Create Subnet

Availability Zone Hints ⓘ

MTU ⓘ

Cancel
« Back
Next »

Obrázek 20: Vytvoření sítě modulu Neutron

12.3.4 Přiřazení sítě k instanci

Cílem sítě je spojovat jednotlivé instance, v rámci jedné či více společných sítí, které mohou být propojeny mezi sebou nebo i k sítím externím, či internetu. Přiřazení sítě se typicky provádí při vytvoření nové instance, tak aby bylo možné spustit či nainicializovat přiřazené síťové rozhraní. Tyto rozhraní jsou poté dostupné u jednotlivých instancí.

testInstance-2

Create Snapshot ▾

Overview
Interfaces
Log
Console
Action Log

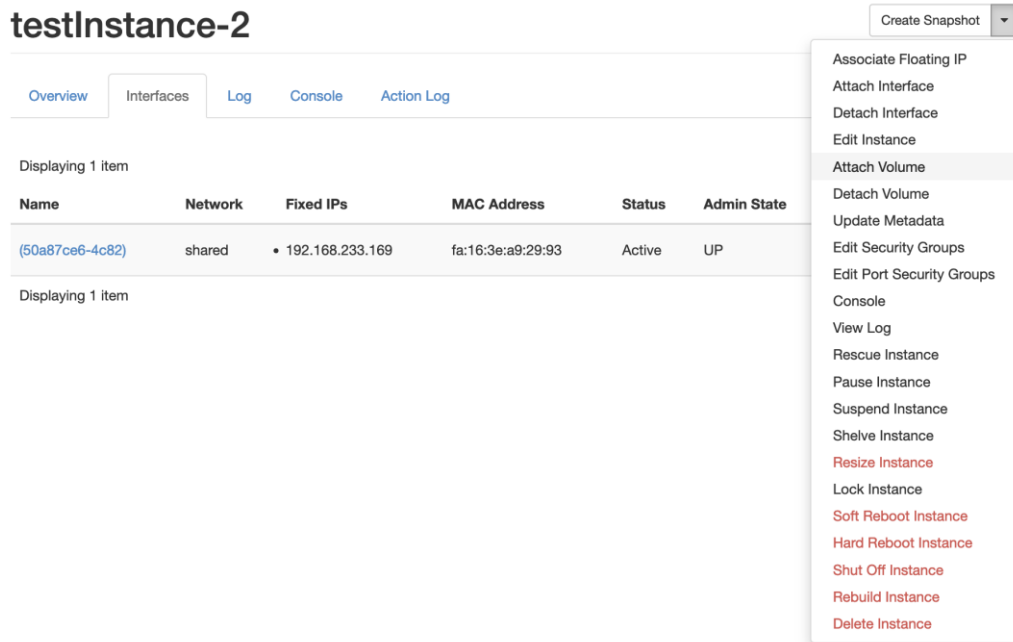
Displaying 1 item

Name	Network	Fixed IPs	MAC Address	Status	Admin State	Actions
(50a87ce6-4c82)	shared	• 192.168.233.169	fa:16:3e:a9:29:93	Active	UP	Edit Security Groups ▾

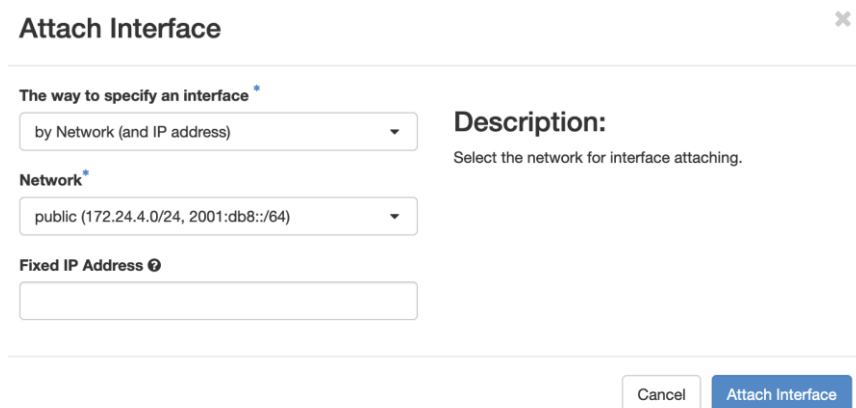
Displaying 1 item

Obrázek 21: Připojená síť modulu Neutron k instanci virtuálního serveru

Po vytvoření instance je možné dodatečně vytvořit nové rozhraní v běžícím operačním systému virtuálního serveru modulu Nova a k tomuto nově vytvářenému rozhraní lze připojit vybranou síť. Tímto je možné je připojit vícero sítí k jednotlivým instancím a zajistit tím vícero spojení a využít tak více adres protokolu IP, pokud jsou potřeba.



Obrázek 22: Kontextová nabídka pro připojení nového rozhraní



Obrázek 23: Přidání nového rozhraní a jeho připojení do sítě modulu Neutron

ZÁVĚR

V této práci byl plnohodnotně vytvořen funkční privátní cloud, s podporou spouštění virtuálních serverů pro potřeby virtualizace či přidělování zdrojů jednotlivým klientům či uživatelům. Takový uzel lze poté použít ke správě velkého množství výpočetních, směrovacích či úložných uzlů napříč rozsáhlým počtem serverů, spojených v jednu ucelenou infrastrukturu, což vede k usnadnění správy prostředků a jejich sjednocenému řízení.

V teoretické části byl popsán software OpenStack, společně s jeho požadavky na hardwarovou výbavu, certifikované operační systémy a další dodatečná softwarová výbava. Blíže k praktické části byl popsán software DevStack, který je souborem instalačních skriptů, jenž umožňují velmi rychle a snáze nainstalovat základní kontrolní uzel platformy OpenStack, s primárním cílem usnadnění vývoje nebo testování platformy OpenStack. Následně byly popsány jednotlivé moduly softwarové platformy OpenStack, včetně jejich bližšího využití a jejich závislostí mezi sebou. Nutno podotknout, že jednotlivé moduly na sobě mohou záviset a pro jejich instalaci a jejich správu je nutné dodržet hierarchii závislostí, tak jak jsou specifikovány v popisu modulů, nebo dle dokumentace.

V rámci praktické části byl sepsán postup, dle kapitol, odpovídající instalaci kompletního ovládacího uzle platformy OpenStack. Nejprve byl nainstalován operační systém Ubuntu Server, konkrétně ve verzi 22.04 LTS. Poté byla doinstalována a nakonfigurována potřebná softwarová výbava, nezbytná pro instalaci platformy OpenStack. Po konfiguraci a ujištění že daný operační systém s nainstalovanou softwarovou výbavou splňuje alespoň minimální požadavky pro provoz softwaru OpenStack, byla platforma nainstalována pomocí instalačních skriptů DevStack. Ve výchozím nastavení byly nainstalovány služby Keystone, Glance, Nova, Placement, Cinder, Neutron a Horizon, kde u bližše zaměřených služeb byla otestována a předvedena funkčnost. Dodatečně, po prvotní instalaci, byl do-instalován modul Swift pro objektové úložiště, u kterého byla taktéž otestována a předvedena funkčnost.

Během zpracovávání této práce se vyskytlo několik problému, jimž jsou věnovány samostatné podkapitoly u modulů, jichž se to přímo týká. Jedná se primárně o chybějící úložiště pro obrazy disků modulu Cinder, nemožnost spuštění virtuálních serverů modulu Nova a problémy s dodatečnou instalací modulu Swift do již sestaveného kontrolního uzle platformy OpenStack.

V současném stavu je server připraven k použití. S kontrolním uzlem platformy OpenStack lze vytvořit vlastní infrastrukturu pomocí dalších uzlů a dle požadavků administrátora začít vytvářet instance virtuálních serverů, nové sítě či nahrávat data na úložný prostor.

POUŽITÁ LITERATURA

- [1] OPENINFRA FOUNDATION. *Openstack*. Online. C2024. Dostupné z: <https://www.openstack.org/>. [cit. 2024-03-12].
- [2] OPENINFRA FOUNDATION. *Documentation for 2023.2 (October 2023)*. Online. 2023. Dostupné z: <https://docs.openstack.org/2023.2/>. [cit. 2024-03-12].
- [3] OPENINFRA FOUNDATION. *Openstack/devstack*. Online. OPENINFRA FOUNDATION. Opendev. C2024. Dostupné z: <https://opendev.org/openstack/devstack>. [cit. 2024-03-12].
- [4] OPENINFRA FOUNDATION. *Openstack/devstack*. Online. GITHUB, INC. GitHub. C2024. Dostupné z: <https://github.com/openstack/devstack>. [cit. 2024-03-12].
- [5] MICROSOFT CORPORATION. *Types of cloud computing*. Online. MICROSOFT CORPORATION. Microsoft Azure. C2024. Dostupné z: <https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/types-of-cloud-computing/>. [cit. 2024-04-27].
- [6] GOOGLE LLC. *What are the different types of cloud computing?* Online. GOOGLE LLC. Google Cloud. C2024. Dostupné z: <https://cloud.google.com/discover/types-of-cloud-computing>. [cit. 2024-04-27].
- [7] OPENINFRA FOUNDATION. *OpenStack wiki*. Online. C2024. Dostupné z: <https://wiki.openstack.org/>. [cit. 2024-03-12].
- [8] OPENINFRA FOUNDATION. *OpenInfra Foundation*. Online. OPENINFRA FOUNDATION. OpenInfra Foundation. C2024. Dostupné z: <https://openinfra.dev/about/>. [cit. 2024-03-12].
- [9] EMMERT, Felix. *Out-of-Band Network Management*. Online. 2015, s. 1-7. Dostupné z: https://doi.org/https://doi.org/doi:%2010.2313/NET-2015-03-1_10. [cit. 2024-03-14].
- [10] INTEL CORPORATION. *What Is Out-of-Band Management (OOBM)?* Online. INTEL CORPORATION. Intel. C2024. Dostupné z: <https://www.intel.com/content/www/us/en/business/enterprise-computers/resources/out-of-band-management.html>. [cit. 2024-03-15].
- [11] NORCINI, Stephanie. *The Definitive Guide To Out-of-Band Management*. Online. OPENGEAR, INC. Opendev. 2021. Dostupné z: <https://opengear.com/blog/the-definitive-guide-to-out-of-band-management/>. [cit. 2024-03-15].
- [12] BURSTEIN, Idan. *Nvidia Data Center Processing Unit (DPU) Architecture*. Online. IEEE. IEEE Xplore. 2021. Dostupné z: <https://ieeexplore.ieee.org/abstract/document/9567066>. [cit. 2024-03-15].
- [13] SDXCENTRAL STUDIOS. *What is NFV (Network Functions Virtualization)? Definition*. Online. SDXCENTRAL STUDIOS. SDxCentral. C2024. Dostupné z: <https://www.sdxcentral.com/networking/nfv/definitions/whats-network-functions-virtualization-nfv/>. [cit. 2024-03-17].

- [14] EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE. *Network Functions Virtualisation (NFV); Architectural Framework*. Online. EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE. ETSI. C2024. Dostupné z: https://www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.01.01_60/gs_NFV002v010101p.pdf. [cit. 2024-03-17].
- [15] HELMKE, Matthew. *Ubuntu Linux: Unleashed*. 2021 Edition. Addison-Wesley Professional, 2020. ISBN 978-0136778851.
- [16] CANONICAL LTD. *Ubuntu Wiki*. Online. C2024. Dostupné z: <https://wiki.ubuntu.com/>. [cit. 2024-03-18].
- [17] *Jammy Jellyfish Release Notes*. Online. ZEMCZAK, Łukasz. CANONICAL LTD. Ubuntu Discourse. 2023. Dostupné z: <https://discourse.ubuntu.com/t/jammy-jellyfish-release-notes/24668>. [cit. 2024-03-18].
- [18] *Ubuntu 22.04.4 LTS (Jammy Jellyfish) Daily Build*. Online. CANONICAL LTD. Ubuntu Releases. C2018. Dostupné z: <https://cdimage.ubuntu.com/jammy/daily-live/current/>. [cit. 2024-03-18].
- [19] *Rufus: Create bootable USB drives the easy way*. Online. BATARD, Pete. Rufus: Create bootable USB drives the easy way. C2011-2024. Dostupné z: <https://rufus.ie/en/>. [cit. 2024-03-20].
- [20] *Rufus: The Reliable USB Formatting Utility*. Online. GITHUB, INC. GitHub. C2024. Dostupné z: <https://github.com/pbatard/rufus>. [cit. 2024-03-20].
- [21] RUBIN, Paul; MACKENZIE, David a KEMP, Stuart. *Dd(1) — Linux manual page*. Online. KERRISK, Michael. Man7.org. 2023. Dostupné z: <https://www.man7.org/linux/man-pages/man1/dd.1.html>. [cit. 2024-03-20].
- [22] OPENINFRA FOUNDATION. *DevStack documentation*. Online. OPENINFRA FOUNDATION. OpenStack Documentation. C2024, 2023-06-07. Dostupné z: <https://docs.openstack.org/devstack/latest/>. [cit. 2024-03-20].
- [23] *Git documentation*. Online. SOFTWARE FREEDOM CONSERVANCY. Git. C2024. Dostupné z: <https://git-scm.com/docs/git>. [cit. 2024-03-20].
- [24] *Useradd(8) — Linux manual page*. Online. KERRISK, Michael. Man7.org. 2023. Dostupné z: <https://www.man7.org/linux/man-pages/man8/useradd.8.html>. [cit. 2024-03-20].
- [25] MACKENZIE, David a MEYERING, Jim. *Chmod(1) — Linux manual page*. Online. KERRISK, Michael. Man7.org. 2023. Dostupné z: <https://www.man7.org/linux/man-pages/man1/chmod.1.html>. [cit. 2024-03-20].
- [26] MILLER, Todd C. *Visudo(8) — Linux manual page*. Online. KERRISK, Michael. Man7.org. 2023. Dostupné z: <https://www.man7.org/linux/man-pages/man8/visudo.8.html>. [cit. 2024-03-20].
- [27] ALLEGRETTO, Chris. *Nano - Nano's ANOther editor, an enhanced free Pico clone*. Online. Linux man page. 1996, 2017. Dostupné z: <https://linux.die.net/man/1/nano>. [cit. 2024-04-12].

- [28] MILLER, Todd C. *Sudo(8) — Linux manual page*. Online. KERRISK, Michael. Man7.org. 2023. Dostupné z: <https://www.man7.org/linux/man-pages/man8/sudo.8.html>. [cit. 2024-03-20].
- [29] MLYNARIK, Richard. *Whoami(1) — Linux manual page*. Online. KERRISK, Michael. Man7.org. 2023. Dostupné z: <https://www.man7.org/linux/man-pages/man1/whoami.1.html>. [cit. 2024-03-20].
- [30] MEYERING, Jim. *Pwd(1) — Linux manual page*. Online. KERRISK, Michael. Man7.org. 2023. Dostupné z: <https://www.man7.org/linux/man-pages/man1/pwd.1.html>. [cit. 2024-03-20].
- [31] STALLMAN, Richard M. a MACKENZIE, David. *Ls(1) — Linux manual page*. Online. KERRISK, Michael. Man7.org. 2023. Dostupné z: <https://www.man7.org/linux/man-pages/man1/lis.1.html>. [cit. 2024-03-23].
- [32] *Cd(1p) — Linux manual page*. Online. KERRISK, Michael. Man7.org. 2023. Dostupné z: <https://www.man7.org/linux/man-pages/man1/cd.1p.html>. [cit. 2024-03-23].
- [33] *Top(1) — Linux manual page*. Online. KERRISK, Michael. Man7.org. 2023. Dostupné z: <https://www.man7.org/linux/man-pages/man1/top.1.html>. [cit. 2024-03-23].
- [34] CHAZARAIN, Guillaume. *Iotop(1) - Linux man page*. Online. Linux man page. 1996, 2017. Dostupné z: <https://linux.die.net/man/1/iotop>. [cit. 2024-03-23].
- [35] OPENINFRA FOUNDATION. *Keystone Installation Tutorial for Ubuntu*. Online. OPENINFRA FOUNDATION. OpenStack Documentation. C2024, 2022-04-22. Dostupné z: <https://docs.openstack.org/keystone/latest/install/index-ubuntu.html>. [cit. 2024-03-27].
- [36] RED HAT, INC. *Managing Services with systemd*. Online. RED HAT, INC. Red Hat Customer Portal. C2024. Dostupné z: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/system_administrators_guide/chap-managing_services_with_systemd. [cit. 2024-03-27].
- [37] FREE SOFTWARE FOUNDATION, INC. *Bash Startup Files*. Online. FREE SOFTWARE FOUNDATION, INC. GNU Operating System. C2024. Dostupné z: https://www.gnu.org/software/bash/manual/html_node/Bash-Startup-Files.html. [cit. 2024-03-27].
- [38] *Export(1p) — Linux manual page*. Online. KERRISK, Michael. Man7.org. 2023. Dostupné z: <https://www.man7.org/linux/man-pages/man1/export.1p.html>. [cit. 2024-03-27].
- [39] SHEPPARD, Simon. . (*source or dot operator*). Online. SHEPPARD, Simon. SS64 Command line reference. C1999-2024. Dostupné z: <https://ss64.com/bash/source.html>. [cit. 2024-03-27].
- [40] OPENINFRA FOUNDATION. *Keystone - Verify operation*. Online. OPENINFRA FOUNDATION. OpenStack Documentation. C2024, 2018-03-31. Dostupné z: <https://docs.openstack.org/keystone/latest/install/keystone-verify-ubuntu.html>. [cit. 2024-03-28].

- [41] OPENINFRA FOUNDATION. *Keystone - Manage projects, users, and roles*. Online. OPENINFRA FOUNDATION. OpenStack Documentation. C2024, 2020-11-25. Dostupné z: <https://docs.openstack.org/keystone/latest/admin/cli-manage-projects-users-and-roles.html>. [cit. 2024-03-29].
- [42] OPENINFRA FOUNDATION. *Horizon - Install and configure for Ubuntu*. Online. OPENINFRA FOUNDATION. OpenStack Documentation. C2024, 2022-01-17. Dostupné z: <https://docs.openstack.org/horizon/latest/install/install-ubuntu.html>. [cit. 2024-04-02].
- [43] *SysVinit*. Online. VINET, Judd; GRIFFIN, Aaron a POLYÁK, Levente. ArchWiki. C2002-2024. Dostupné z: <https://wiki.archlinux.org/title/SysVinit>. [cit. 2024-04-02].
- [44] OPENINFRA FOUNDATION. *Nova - Install and configure controller node for Ubuntu*. Online. OPENINFRA FOUNDATION. OpenStack Documentation. C2024, 2023-02-15. Dostupné z: <https://docs.openstack.org/nova/latest/install/controller-install-ubuntu.html>. [cit. 2024-04-06].
- [45] OPENINFRA FOUNDATION. *Nova - Install and configure a compute node for Ubuntu*. Online. OPENINFRA FOUNDATION. OpenStack Documentation. C2024, 2023-02-15. Dostupné z: <https://docs.openstack.org/nova/latest/install/compute-install-ubuntu.html>. [cit. 2024-04-06].
- [46] OPENINFRA FOUNDATION. *Nova - Host aggregates*. Online. OPENINFRA FOUNDATION. OpenStack Documentation. C2024, 2020-08-04. Dostupné z: <https://docs.openstack.org/nova/latest/admin/aggregates.html>. [cit. 2024-04-11].
- [47] OPENINFRA FOUNDATION. *Glance Installation*. Online. OPENINFRA FOUNDATION. OpenStack Documentation. C2024, 2019-07-03. Dostupné z: <https://docs.openstack.org/glance/latest/install/index.html>. [cit. 2024-04-12].
- [48] OPENINFRA FOUNDATION. *DevStack documentation - Swift*. Online. OPENINFRA FOUNDATION. OpenStack Documentation. C2024, 2022-11-14. Dostupné z: <https://docs.openstack.org/devstack/latest/configuration.html#swift>. [cit. 2024-04-26].
- [49] OPENINFRA FOUNDATION. *Neutron - Install and configure controller node*. Online. OPENINFRA FOUNDATION. OpenStack Documentation. C2024, 2023-01-03. Dostupné z: <https://docs.openstack.org/neutron/latest/install/controller-install-ubuntu.html>. [cit. 2024-04-24].
- [50] OPENINFRA FOUNDATION. *Neutron - Install and configure compute node*. Online. OPENINFRA FOUNDATION. OpenStack Documentation. C2024, 2023-10-25. Dostupné z: <https://docs.openstack.org/neutron/latest/install/compute-install-ubuntu.html>. [cit. 2024-04-24].

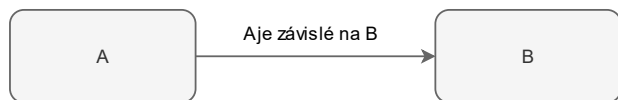
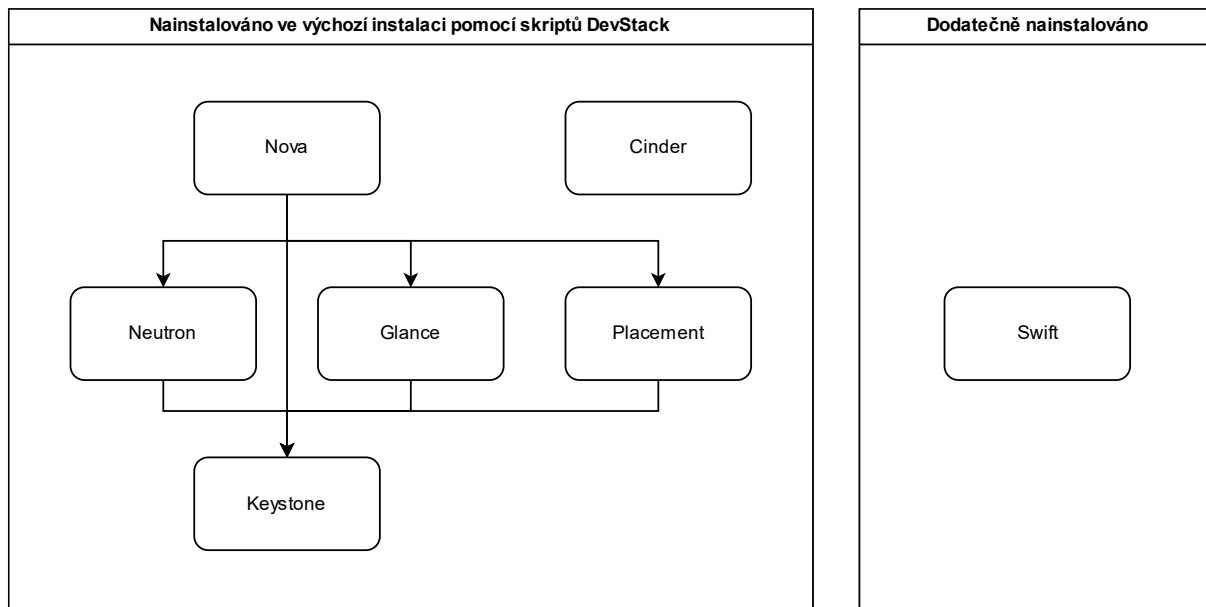
SEZNAM PŘÍLOH

Příloha A: Diagram nainstalovaných modulů.

Příloha B: Logy příkazů spuštěných na serveru.

PŘÍLOHA A: Diagram nainstalovaných modulů

Tento diagram ukazuje nainstalované moduly platformy OpenStack. Rozdělené na výchozí instalaci pomocí instalačních skriptů DevStack a dodatečně nainstalovaný modul Swift.



PŘÍLOHA B: Logy příkazů spuštěných na serveru

Tento soubor ZIP obsahuje kolekci logů ze vzdáleného přístupu k serveru pomocí protokolu SSH. Obsaženy jsou veškeré podstatné příkazy pro úspěšnou instalaci platformy OpenStack a její následovnou správu, tak jak byly tyto příkazy použity na reálném serveru.