PROJEKT

Multitenantné operačné centrum kybernetickej bezpečnosti riešené
ako otvorená cloudová služba s prvkami strojového učenia

V3 popis vybraného riešenia spĺňajúceho stanovené kritériá

KPB1 – Návrh virt. riešenia

typ – dokumentácia

mesiac odovzdania – M7

Financované
Európskou úniou
NextGenerationEU

PLÁN OBNOVY

ÚRAD VLÁDY
SLOVENSKEJ REPUBLIKY

Obsah

1. Cloud Computing 5
1.1. Základné charakteristiky CC 5
1.2. Model poskytovania služieb: Softvér ako Služba (SaaS) 5
1.2.1. Kto sú konzumenti? 5
1.2.2. Čo dostane konzument? 5
1.2.3. Ako sa počítajú poplatky? 5
1.2.4. Vzťah k vrstvovému modelu 6
1.2.5. Príklady vhodného použitia 6
1.2.6. Príklady nevhodného použitia 6
1.2.7. Výhody 6
1.3. Model poskytovania služieb: Platforma ako Služba (PaaS) 7
1.3.1. Kto sú konzumenti? 7
1.3.2. Čo dostane konzument? 7
1.3.3. Ako sa počítajú poplatky? 7
1.3.4. Vzťah k vrstvovému modelu 7
1.3.5. Príklady použitia 8
1.3.6. Výhody 8
1.4. Model poskytovania služieb: Infraštruktúra ako Služba (IaaS) 8
1.4.1. Kto sú konzumenti? 8
1.4.2. Čo dostane konzument? 8
1.4.3. Ako sa počítajú poplatky? 8
1.4.4. Vzťah k vrstvovému modelu 8
2. Systém OpenStack 10
2.1. Oboznámenie sa s modulami OpenStacku 10
2.1.1. KeyStone 10
2.1.2. Nova 10
2.1.3. Neutron 10
2.1.4. Horizon 10
2.1.5. Skyline 11
2.1.6. Glance 11
2.1.7. Swift 11
2.1.8. Cinder 11
2.1.9. Manila 11
2.2. Siete v OpenStack – modul Neutron 11
2.2.1. Prepínanie 11
2.2.2. Smerovanie 12
2.2.3. Load balancing 12
2.2.4. Firewalling 12
2.3. Sieťová architektúra OpenStack cloudu 12
2.3.1. Sieťové zóny 12
2.3.2. Možnosti sieťovej vrstvy 13
2.4. Typy sieti 14

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
Financované Európskou úniou NextGenerationEU

2.5. Sieť nájomcov (Tenant network) 15
2.6. Plugin Modular Layer 2 (ML2) 15
2.7. OpenVSwitch (OVS) 15
2.8. Uzly služby OpenStack 16
2.8.1. Riadiaci uzol (Controller node) 16
2.8.2. Výpočtový uzol (Compute node) 16
2.9. Vysoká dostupnosť systému OpenStack 16
2.9.1. RabbitMQ 16
2.9.2. MariaDB – Galera klaster 16
2.9.3. Keepalived 17
3. Automatizované produkčné nasadenie platformy OpenStack 18
3.1. Nástroje na automatizované nasadenie OpenStacku 19
3.2. OpenStack Charms 20
3.2.1. Metal-as-a-Service(MAAS) 20
3.3. Komponent Juju 21
4. Platforma pre správu cloudu 22
4.1. ManageIQ 22
4.1.1. Architektúra ManageIQ 23
4.1.2. Požiadavky na platformu 24
4.2. Centralizovaná autentifikácia ManageIQ 25
4.2.1. Konfigurácia autentifikácie v prostredí ManageIQ 25
4.2.2. Inštalácia a konfigurácia SSSD 26
4.2.3. Testovanie konfigurácie 26
4.2.4. Integrácia s Apache a ManageIQ UI 26
5. Distribuované úložisko 27
5.1. Princípy a výhody distribuovaného úložiska 27
5.2. Využitie distribuovaného úložiska 28
5.3. Dôvody nasadenia distribuovaného úložiska 28
6. Výber softvéru distribuovaného úložiska pre OpenStack 29
6.1. Kritéria pre výber softvéru 29
6.2. Dostupné možnosti 30
6.3. Výber softvéru 30
7. Ceph A jeho architektúra 32
7.1. Kľúčové komponenty softvéru Ceph 32
7.1.1. Ceph klaster 32
7.1.2. Ceph Monitor 33
7.1.3. Ceph Manager 34
7.1.4. OSD démony 34
7.1.5. MDS a Ceph File System 35
7.2. RADOS 36
7.3. Algoritmus CRUSH 37
7.4. Pool-y a PGs 38
7.4.1. Pool 39
7.4.2. PG 39

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
Financované Európskou úniou NextGenerationEU

8. IMPLEMENTÁCIA ZDIEĽANÉHO SÚBOROVÉHO SYSTÉMU V PROSTREDÍ OPENSTACK 41
8.1. Úložiská v prostredí platformy OpenStack 41
8.2. Manila ako služba pre správu zdieľaných sieťových súborových systémov v prostredí OpenStack 41
8.3. Ciele a výhody použitia služby Manila 43
8.4. Kľúčové komponenty architektúry služby Manila 43
8.5. Interakcia komponentov služby Manila 44
9. Monitoring v prostredí Cloud computing-u 46
9.1. Motivácia pre monitorovanie 46
9.1.1. Kapacita a plánovanie zdrojov 46
9.1.2. Kapacita a manažovanie zdrojov 46
9.1.3. Manažovanie dátového centra 47
9.1.4. Manažment SLA 47
9.1.5. Fakturovanie 47
9.1.6. Zisťovanie problémov 47
9.1.7. Manažovanie výkonu 47
9.1.8. Bezpečnosť 48
9.2. Pohľady na monitoring 48
Zoznam zdrojov 50

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
Financované Európskou úniou NextGenerationEU

1. CLOUD COMPUTING

Dokument [26] definuje cloud computing (CC) ako model systému, v ktorom poskytovateľ umožňuje konzumentovi pohodlný všadeprítomný prístup k zdieľanej konfigurovateľnej výpočtovej sile (siete, servery, úložisko, aplikácie a služby) na požiadanie. Táto výpočtová sila môže byť poskytnutá alebo uvoľnená s minimálnym zásahom poskytovateľa.

1.1. Základné charakteristiky CC

Dokument [26] bližšie definuje základné charakteristiky CC:

- **Samoobslužná interakcia na požiadanie:** konzumentovi môže byť automaticky poskytnutá výpočtová sila, ktorú potrebuje bez ľudskej interakcie s poskytovateľom služby.
- **Vysoká dostupnosť siete:** prostriedky sú prístupné cez sieť a dá sa k nim pristupovať z rôznych štandardných zariadení ako je napr. PC, mobilné telefóny a tablety.
- **Zdieľanie prostriedkov:** výpočtová sila poskytovateľa je zdieľaná medzi viacerých konzumentov. Fyzické a virtuálne prostriedky sú dynamicky prideľované a odoberané podľa ich požiadaviek. Takto konzument nepozná presné miesto, kde sa jemu pridelené prostriedky nachádzajú (pri zovšeobecnení sa lokalita dá zúžiť napr. na štát alebo dátové centrum). Príkladom takýchto prostriedkov sú úložný priestor, CPU, pamäť a šírka pásma siete.
- **Rýchla elasticita:** prostriedky sú elasticky prideľované a odoberané, v niektorých prípadoch automaticky, aby sa rýchlo prispôsobili odpovedajúcim požiadavkám. Pre konzumenta sa tak prostriedky javia ako neobmedzené, ktoré môže využívať v hocijakom množstve, v hocijakom čase.
- **Meraná služba:** Cloud systém automaticky kontroluje a optimalizuje prostriedky, tak že implementuje meranie v niektorej úrovni abstrakcie, potrebnej pre daný druh monitorovania (napr. úložný priestor alebo aktívne používateľské účty). Využitie prostriedkov môže byť monitorované, kontrolované a zhrnuté vo výstupoch, čím poskytuje transparentnosť pre poskytovateľa aj pre konzumenta danej služby.

1.2. Model poskytovania služieb: Softvér ako Služba (SaaS)

Službou pre konzumenta sú aplikácie prístupné cez internet, sprostredkovávané poskytovateľom. Prístupné sú rôznym zariadeniam konzumenta napr. webovým prehliadačom alebo programovým rozhraním.

1.2.1. Kto sú konzumenti?

- Organizácie, ktoré poskytujú pre zamestnancov prístup k typickým kancelárskym aplikáciám (tabuľkový procesor, email a pod.).
- Koncový používatelia, ktorí priamo používajú poskytovaný softvér.
- Administrátori aplikácií, ktorí ich konfigurujú pre koncových zákazníkov.

1.2.2. Čo dostane konzument?

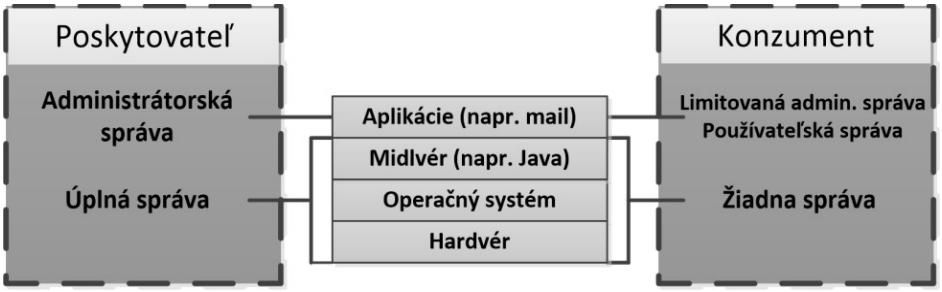
Právo používať konkrétnu aplikáciu na požiadanie a správu dát ako je ich zálohovanie a zdieľanie medzi konzumentmi.

1.2.3. Ako sa počítajú poplatky?

Zvyčajne sú založené na počte používateľov, času používania aplikácie, počtu použitia, spustenia, spracovaných záznamov; použitej šírky pásma a množstva uložených dát.

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
Financované Európskou úniou NextGenerationEU

1.2.4. Vzťah k vrstvovému modelu



Obr. 7 - SaaS – vzťah k vrstvovému modelu

Konzument platí za prístup k aplikácii, avšak neriadi ani neovláda spodné vrstvy modelu (sieť, servery, operačný systém, úložný priestor, ani niektoré nastavanie využívanej aplikácie):

V modeli SaaS má konzument prístup k prostriedkom, ktoré mu pridelí SaaS aplikácia poskytovateľa. Napríklad email, kde konzument dokáže vytvárať, posielať a ukladať poštu. Ak to aplikácia umožňuje, tak má možnosť obmedzenej administrátorskej kontroly (vytváranie účtov pre iných konzumentov, prehľad aktivít a pod.).

Poskytovateľ je administrátorom so všetkými právami SaaS aplikácie. Jeho úlohou je konfigurácia, aktualizácia a poskytovanie požadovaných služieb.

1.2.5. Príklady vhodného použitia

- **obchodná činnosť –** prepojenie obchodníkov s dodávateľmi alebo zákazníkmi (mail, fakturácia, riadenie zásob),
- **spolupráca –** aplikácie pre tímovú spoluprácu (zdieľanie obrazovky, dokumentov; kalendár, online hranie hier),
- **produktivita kancelárie –** aplikácie ako textové editory, tabuľkové procesory, programy na prezentácie a databázy (doplňujú programy o možnosti spolupráce),
- **softvérové nástroje –** aplikácie pre správu bezpečnosti, aplikácie pre tvorbu webu a pod.

1.2.6. Príklady nevhodného použitia

- **softvér pracujúci v reálnom čase –** aplikácie pre letectvo, ovládanie priemyselných robotov a pod. (prenos dát cez Cloud nedokáže zaručiť dostatočne rýchlu odozvu),
- **prenos obrovských dát –** aplikácie v medicíne alebo fyzike, kde sa generuje obrovské množstvo dát, čo nemusí byť výhodné prenášať cez Cloud,
- **kritický softvér –** aplikácie kde chyba môže znamenať poškodenie majetku, alebo stratu života (problém s odozvou).

1.2.7. Výhody

- **Jednoduchosť –** webový prehliadač konzumenta je dostatočne efektívny na zobrazenie požadovaných dát a tak nie je nutnosť ďalších aplikácií. Nie je preto nutná žiadna inštalácia softvéru, v PC konzumenta sa tak nachádza minimum dát súvisiacich s aplikáciou.
- **Efektívne využitie licencií –** jedna licencia môže byť naraz použitá pre viacerých konzumentov. Je to umožnené tým, že softvér je na strane poskytovateľa a využívanie licencovaných produktov konzumenta si poskytovateľ zahŕňa v cene poskytovania služieb.

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
Financované Európskou úniou NextGenerationEU

- **Centralizovaná správa a dáta –** z pohľadu konzumenta sa javia všetky dáta ako centralizované nakoľko sa nachádzajú u poskytovateľa (v skutočnosti sú decentralizované kvôli redundancii a pod). Logická centralizácia zaručuje konzumentovi ochranu pred stratou dát, bezpečnosť, umožňuje robiť zálohovanie a pod.
- **Zodpovednosť za platformu preberá poskytovateľ –** konzument sa nezúčastňuje správy nižších vrstiev.
- **Úspora financií –** poskytovateľ poskytuje všetok hardvér, napájanie, údržbu zariadení atď., čo by malo v konečnom dôsledku konzumenta stáť menej finančných prostriedkov (aj za cenu poplatkov), ako by si to mal zabezpečovať sám.

1.3. Model poskytovania služieb: Platforma ako Služba (PaaS)

Službou pre konzumenta je možnosť umiestniť vlastný softvér do prostredia cloud alebo možnosť využiť programovacie jazyky, knižnice a nástroje poskytnuté poskytovateľom.

1.3.1. Kto sú konzumenti?

- Vývojári aplikácií.
- Aplikačný testeri, ktorí testujú aplikáciu v rôznych prostrediach.
- Programátori, ktorí zavádzajú nové aplikácie (alebo ich aktualizácie) do prostredia cloud.
- Administrátori, ktorí konfigurujú, ladia a monitorujú výkon aplikácií.
- Koncový používatelia, ktorí používajú aplikáciu (pre nich je prístup k aplikácii rovnaký ako pri modely SaaS).

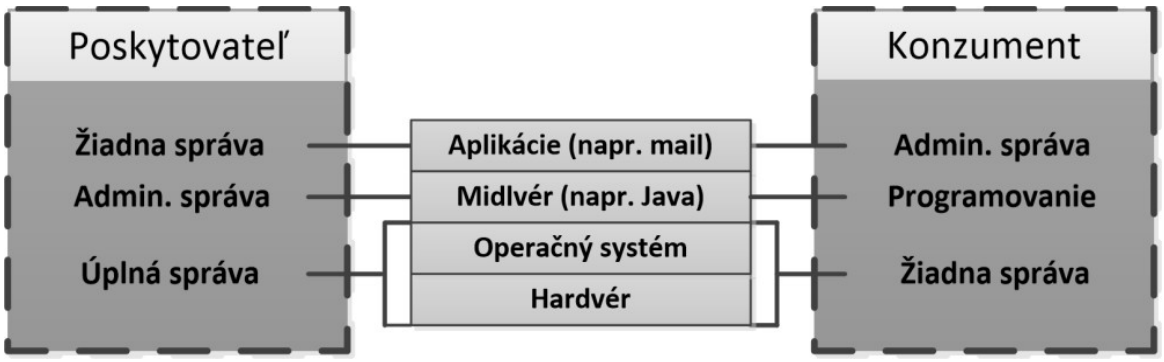
1.3.2. Čo dostane konzument?

Možnosť použiť poskytovateľove nástroje a prostriedky na beh aplikácií, vývoj, testovanie, zavedenie a administráciu.

1.3.3. Ako sa počítajú poplatky?

Zvyčajne sú založené na počte používateľov, ich type (napr. vývojár/koncový používateľ), veľkosti úložiska, použitej výpočtovej sile, šírke pásma, ktoré platforma použila a dĺžke času jej používania.

1.3.4. Vzťah k vrstvovému modelu



Obr. 8 - PaaS – vzťah k vrstvovému modelu

Konzument neriadi ani neovláda spodné vrstvy modelu (sieť, servery, operačný systém, úložný priestor, ale má možnosť ovládať a konfigurovať aplikácie).

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
Financované Európskou úniou NextGenerationEU

V modeli PaaS riadi najspodnejšie vrstvy modelu poskytovateľ (operačný systém, hardvér, sieť v LAN a medzi dátovými centrami). Konzumentovi poskytuje prostredie na beh a programovanie jeho aplikácií a tiež výkon CPU, pamäť, úložný priestor, databázy, sieťové pripojenie a pod. Poskytovateľ určuje programovací model t.j. okolnosti, za ktorých bude môcť konzument programovať vlastné aplikácie (za účelom spoplatnenia). Konzument má plnú moc nad vlastnými aplikáciami.

Model PaaS poskytuje priestor pre vývoj, beh a administráciu softvéru, ktorý je určený pre veľké množstvo konzumentov, spracováva veľké množstvo dát a potencionálne bude dosiahnuteľný z ktoréhokoľvek miesta cez internet.

PaaS je veľmi podobný tradičnému počítačovému systému. Na rozdiel od neho aplikácie v modeli PaaS môžu využiť toľko výpočtovej sily koľko potrebujú, spracovať veľké množstvo dát, byť zavedené skoro okamžite a obsluhovať veľké množstvo používateľov.

1.3.5. Príklady použitia

Prostredie modelu PaaS môže byť využité na vývoj aplikácií poskytujúcich služby spomenutých pri modely SaaS.

1.3.6. Výhody

Konzument je oslobodený od povinnosti výberu, inštalácie, údržby alebo ovládania komponentov platformy. Veľa výhod ma spoločných s modelom SaaS (*jednoduchosť, centralizovaná správa a dáta, zodpovednosť za platformu preberá poskytovateľ, úspora financií*).

1.4. Model poskytovania služieb: Infraštruktúra ako Služba (IaaS)

Službou pre konzumenta je poskytnutie výpočtovej sily, úložného priestoru, siete a ostatných základných prostriedkov tak, aby mohol umiestniť a spustiť ľubovoľný softvér v systéme cloud (vrátane operačného systému a aplikácií).

1.4.1. Kto sú konzumenti?

Systémoví administrátori.

1.4.2. Čo dostane konzument?

Prístup k virtuálnemu počítaču, úložný priestor dostupný cez sieť, sieťové prvky ako je firewall a možnosť konfigurácie služieb.

1.4.3. Ako sa počítajú poplatky?

Zvyčajne využitie CPU/hodinu, uložené dáta/hodinu, spotrebovaná šírka pásma, využitie sieťovej infraštruktúry (napr. IP adresy)/hodinu a služby s prídavnou hodnotou (napr. monitoring, automatická škálovateľnosť a pod.).

1.4.4. Vzťah k vrstvovému modelu

Pri modeli IaaS je vrstva operačné systému rozdelená na dve časti. Pridáva sa nižšia (a viac privilegovaná) vrstva, kde je spustený Virtual Machine Monitor (VMM), ktorý sa nazýva hypervisor. Používa sa na spoluprácu viacerých Virtual Machines (VM), kde sa VM konzumentovi javí ako fyzický počítač, ktorý sa dá konfigurovať cez sieť. Operačný systém, ktorý je nainštalovanývo VM sa nazýva hosťovský operačný systém.

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
Financované Európskou úniou NextGenerationEU



Obr. 9 - IaaS – vzťah k vrstvovému modelu

Poskytovateľ ovláda najnižšie vrstvy modelu (fyzický hardvér a hypervisor). Konzument môže vytvoriť požiadavku na vytvorenie novej VM. Tejto požiadavke je vyhovené, len ak spĺňa politiku poskytovateľa. Cez hypervisor môže poskytovateľ poskytnúť konzumentovi virtuálne sieťové prvky, tak aby si mohol vytvoriť a nakonfigurovať vlastnú virtuálnu sieť v rámci poskytovateľovej infraštruktúry. Úlohou poskytovateľa však naďalej ostáva zodpovednosť za riadenie, aktualizáciu a konfiguráciu infraštruktúry za účelom bezpečnosti a spoľahlivosti.

Konzument má v modely IaaS úplnú kontrolu nad hosťovským operačným systémom v každom VM a nad všetkými vrstvami nad ním.

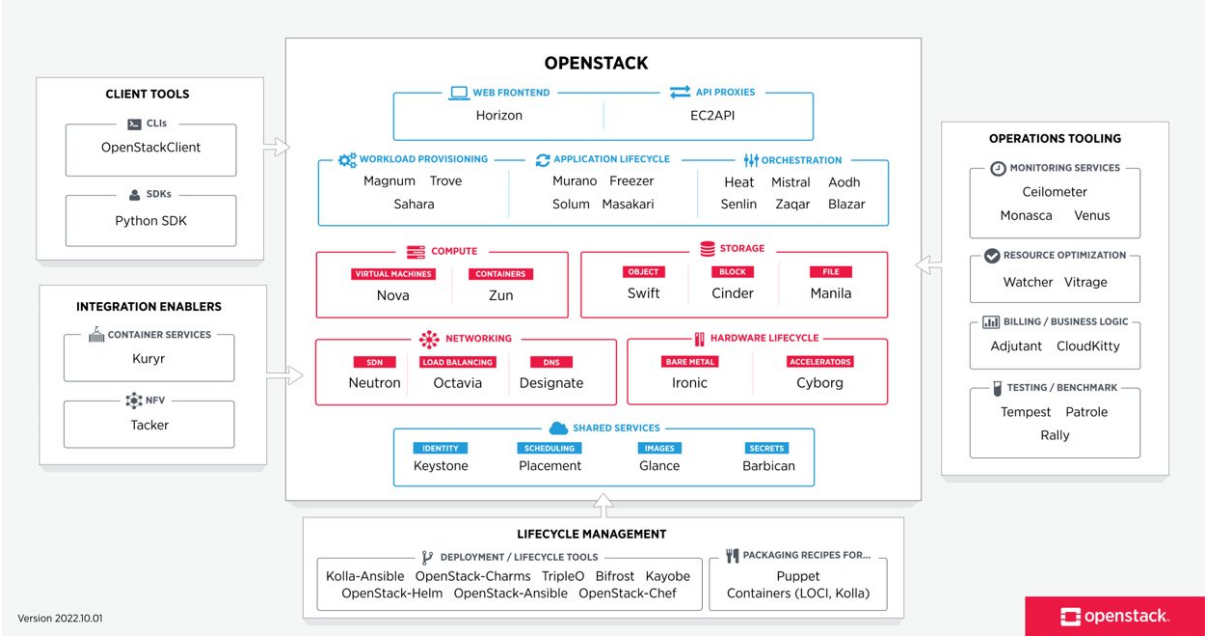
Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
Financované Európskou úniou NextGenerationEU

2. SYSTÉM OPENSTACK

2.1. Oboznámenie sa s modulami OpenStacku

OpenStack je cloudový operačný systém, ktorý funguje pomocou modulov. Každý modul má pridelenú funkciu alebo oblasť ktorú spravuje. Tieto moduly medzi sebou komunikujú pomocou API rozhraní. Približne každý polrok vzniká nová verzia OpenStacku. S každou novou verziou množstvo týchto modulov pribúda, zaniká alebo sa upraví/rozšíri ich funkcionalita. V najnovšej verzii (2025.1) je týchto modulov okolo 60.

Medzi najdôležitejšie moduly, nevyhnutné pre fungovanie OpenStacku patria: KeyStone, Nova, Neutron, Horizon (alebo v najnovšej verzii Skyline), Glance, Swift, Cinder, Manila.



Obrázok 2.1 Zdroj: (Techtarget, 2025)

2.1.1. KeyStone

Keystone je modul OpenStacku, ktorý poskytuje autentifikáciu klienta pomocou API, zisťovanie služieb a distribuovanú autorizáciu viacerých klientov, implementáciou rozhrania OpenStack Identity API. Podporuje LDAP, OAuth, OpenID Connect, SAML a SQL.

2.1.2. Nova

Modul Nova implementuje služby a súvisiace knižnice, ktoré poskytujú masívne škálovateľný, na požiadanie samoobslužný prístup k výpočtovým zdrojom, vrátane holého kovu (bare-metal), virtuálnych strojov a kontajnerov.

2.1.3. Neutron

OpenStack Neutron je sieťový projekt SDN (Software-Defined Networking) zameraný na poskytovanie networking-as-a-service (NaaS) vo virtuálnych výpočtových prostrediach.

2.1.4. Horizon

Horizon poskytuje službu dashboardu – webového rozhrania OpenStacku, ktorý je rozšíriteľný a poskytuje webové používateľské rozhranie pre jednotlivé služby OpenStacku.

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
Financované Európskou úniou NextGenerationEU

2.1.5. Skyline

Skyline je optimalizovaný dashboard OpenStacku. Má moderné a technologicky vyspelé prostredie. Pre vývojárov je ľahšie udržiavateľný a prevádzkovaný používateľmi. Poskytuje vyšší výkon súbežnosti.

2.1.6. Glance

Glance je modul, ktorý spravuje manažment obrazov (imagov) virtuálnych zariadení (VM). V rámci manažmentu obrazov VM je možné hľadať a ukladať na rôznych miestach od jednoduchých súborových systémov až po systémy na ukladanie objektov, ako je modul OpenStacku – Swift.

2.1.7. Swift

Projekt OpenStack Object Store, známy ako Swift, ponúka cloud storage softvér, takže je možné ukladať a načítavať veľké množstvo údajov pomocou jednoduchého rozhrania API. Je vytvorený a optimalizovaný pre odolnosť, dostupnosť a súbežnosť v celom súbore údajov. Swift je ideálny na ukladanie neštruktúrovaných údajov, ktoré môžu rásť bez hraníc.

2.1.8. Cinder

Modul Cinder je určený pre správu blokového úložiska OpenStacku. Virtualizuje správu blokových úložných zariadení a poskytuje koncovým používateľom samoobslužné API na vyžiadanie a spotrebu týchto zdrojov bez toho, aby vyžadovali akékoľvek znalosti o tom, kde je ich úložisko skutočne nasadené alebo na akom type zariadenia. To sa deje pomocou referenčnej implementácie LVM (Logical Volume Management) alebo ovládačov doplnkov pre iné úložisko.

2.1.9. Manila

Manila je modul, ktorý poskytuje koordinovaný prístup k zdieľaným alebo distribuovaným súborovým systémom.

2.2. Siete v OpenStack – modul Neutron

OpenStack networking je škálovateľný systém založený na API, umožňujúci spravovať siete a IP adresy v OpenStack cloude. Rovnako ako iné kľúčové komponenty dokáže byť nástrojom pre administrátorov a používateľov na maximalizáciu využitia existujúcich fyzických zdrojov cloudovej infraštruktúry. Neutron je komponent, ktorý implementuje riešenie SDN. Poskytuje nájomníkom úplnú kontrolu nad vytváraním prostriedkov virtuálnej siete. Toho sa často dosahuje formoutunelovacích protokolov, ktoré vytvárajú zapuzdrené komunikačné cesty cez existujúcu sieťovú infraštruktúru s cieľom segmentovať prenosy nájomníkov. Táto metóda sa líši v závislosti od konkrétnej implementácie, ale niektoré z najbežnejších metód zahŕňajú tunelovanie cez GRE, zapuzdrovanie pomocou značiek VXLAN a VLAN [4].

Neutron je samostatná služba, ktorú je možné nainštalovať nezávisle od ostatných služieb. Obsahuje mnoho technológií, ktoré by sa dali nájsť v dátovom centre vrátane prepínania, smerovania, load balancing, firewallingu a virtuálnych privátnych sietí.

2.2.1. Prepínanie

Virtuálny prepínač je definovaný ako softvérová aplikácia, ktorá spája virtuálne stroje s virtuálnymi sieťami na druhej vrstve alebo data-linkovej vrstve OSI modelu. Neutron podporuje viacej platforiem prepínanie vrátane Linux bridges, poskytovaný bridge modulom kernelu, ale aj známy softvérový prepínač OVS (Open vSwitch). OVS je open source virtuálny prepínač, ktorý podporuje štandardné rozhrania a protokoly ako NetFlow, SPAN, RSPAN, LACP a 802.1q VLAN tagovanie. Okrem VLAN tagovania môžu používatelia vytvárať overlay siete pomocou tunelovacích protokolov L2-in-L3, ako sú GRE alebo VXLAN. OVS možno

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
Financované Európskou úniou NextGenerationEU

použiť na uľahčenie komunikácie medzi inštanciami a zariadeniami mimo OpenStack, ktoré zahŕňajú hardvérové prepínače, firewally, úložné zariadenia a dedikované servery [3].

2.2.2. Smerovanie

Neutron poskytuje možnosti smerovania a NAT pomocou využívania IP forwardingu, iptables a sieťových namespaces. V sieťovom namespace môžeme nájsť sokety, viazané porty a rozhrania, ktoré boli v tomto namespace vytvorené. Každý namespace má svoju vlastnú smerovaciu tabuľku, rozhrania, a iptables ktoré poskytujú filtrovanie a preklad sieťových adries. Vďaka rozdeleniu sieti do rôznych namespaces nevznikajú obavy z prekrývania podsieti medzi rôznymi používateľmi. Konfigurácia smerovača v rámci Neutronu umožňuje inštanciám komunikovať medzi cloudovými vonkajšími sieťami. Namespaces využívajú aj pokročilé sieťové služby ako „Firewall as a service“ a „Virtual Private Network as a service“ (Denton, 2015).

2.2.3. Load balancing

Load balancing poskytuje používateľom schopnosť distribuovať požiadavky klientov naprieč viacerými inštanciami alebo servermi. Používatelia môžu vytvárať monitorovanie, nastavovať limity pripojenia a aplikovať profily perzistencie na prevádzku prechádzajúcu cez virtuálny load balancer [3].

2.2.4. Firewalling

OpenStack ponúka dva spôsoby na zabezpečenie inštancií: security groups a firewalls. Pri použití security groups sa inštancie umiestňujú do skupín, ktoré zdieľajú spoločné funkcie a sady pravidiel. Pravidlá v Iptables sú konfigurované na výpočtových uzloch a filtrujú prenos do a z Linux bridges pripojených ku každej inštancii. Pri používaní virtuálneho firewallu ako FWaaS, je prevádzka filtrovaná na okraji siete na Neutron smerovači a nie na výpočtovom uzle [3].

Virtuálne privátne siete alebo VPN, rozširujú privátnu sieť cez verejnú sieť ako napríklad Internet. VPN umožňuje počítaču odosielať a prijímať údaje cez verejné siete, akoby bol priamo pripojený k súkromnej sieti. Neutron poskytuje sadu rozhraní API, ktoré umožňujú používateľom vytvárať VPN tunely založené na protokole IPsec zo smerovačov na vzdialené brány [4].

2.3. Sieťová architektúra OpenStack cloudu

OpenStack poskytuje bohaté sieťové prostredie. V tejto časti si popíšeme požiadavky a možnosti, ktoré je potrebné zohľadniť pri navrhovaní privátneho cloudu. Patria sem odporúčania sieťových implementácií, ktoré je potrebné zohľadniť, informácie o niektorých sieťových usporiadaniach OpenStack a sieťových službách, ktoré sú nevyhnutné pre stabilnú prevádzku. Cloudové prostredie zásadným spôsobom mení spôsoby poskytovania a využívania sietí. Pri rozhodovaní o architektúre je nevyhnutné porozumieť nasledujúcim konceptom a rozhodnutiam.

2.3.1. Sieťové zóny

Cloudové siete sú rozdelené do niekoľkých logických zón. Je odporúčané definovať najmenej štyri odlišné sieťové zóny.

OpenStack Underlay network – Underlay je definovaná ako fyzická sieťová prepínacia infraštruktúra, ktorá spája úložné, výpočtové a riadiace platformy. Existuje veľké množstvo potenciálnych možností underlay siete.

OpenStack Overlay network – Overlay je definovaný ako akákoľvek konektivita L3 medzi cloudovými komponentmi a môže mať formu riešení SDN, ako je riešenie neutron overlay alebo riešenia SDN tretích strán.

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
Financované Európskou úniou NextGenerationEU

Edge – Edge zóna je miesto, kde sieťová prevádzka prechádza z cloudového overlay alebo sietí SDN do tradičných sieťových prostredí.

External – External zóna je definovaná ako konfigurácia a komponenty, ktoré sú potrebné na zabezpečenie prístupu k cloudovým zdrojom. External je definovaná ako všetky komponenty mimo cloudových okrajových brán [4].

Tok dát

V rámci cloudovej infraštruktúry existujú dva primárne typy dopravného toku, výber sieťových technológií je ovplyvnený očakávaným zaťažením cloudu:

- **East/West** – Vnútorný tok dát medzi virtuálnymi zariadeniami v cloude a tok dát medzi výpočtovými uzlami a uzlami úložiska. Spravidla ide o najväčší tok a z dôvodu potreby postarať sa o úložisko je potrebné zabezpečiť minimum hopov a nízku latenciu.
- **North/South** – Tok dát medzi virtuálnymi zariadeniami a všetkými externými sieťami vrátane klientov a vzdialených služieb. Tento tok dát veľmi závisí od pracovnej záťaže v cloude a od typu ponúkaných sieťových služieb [4]

2.3.2. Možnosti sieťovej vrstvy

Pri rozhodovaní o tom, či použiť sieťovú architektúru vrstvy 2 alebo sieťovú architektúru vrstvy 3, je potrebné vziať do úvahy niekoľko faktorov.

Benefity L2 vrstvy

Existuje niekoľko dôvodov, prečo sa sieť navrhnutá na protokoloch L2 uprednostňuje pred L3 sieťou.

- Ethernetové rámce obsahujú všetko podstatné pre prácu v sieti. Patria sem okrem iného globálne jedinečné zdrojové adresy, globálne jedinečné cieľové adresy a kontrola chýb.
- Ethernetové rámce môžu niesť akýkoľvek druh paketu. Sieť na vrstve 2 je nezávislá od protokolu vrstvy 3.
- Do siete Ethernet môžete pridávať doplnkové sieťové funkcie, napríklad triedu služby (CoS) alebo multicasting, rovnako ľahko ako siete IP.
- VLAN sú ľahkým mechanizmom na izoláciu sietí.
- Rýchlosť
- Znížená réžia hierarchie IP.
- Pri pohybe systémov nie je potrebné sledovať či meniť konfiguráciu adries.

Zatiaľ čo jednoduchosť protokolov L2 môže fungovať dobre v dátovom centre so stovkami fyzických strojov, cloudové dátové centrá majú ďalšiu záťaž spočívajúcu v nutnosti sledovať všetky adresy a siete virtuálnych strojov. V týchto dátových centrách nie je nezvyčajné, že jeden fyzický uzol podporuje 30 až 40 virtuálnych inštancií [4].

Limitácie L2 vrstvy

Sieťové architektúry L2 vrstvy majú určité obmedzenia, ktoré sú citeľné pri použití mimo tradičných dátových centier.

- Počet VLAN je obmedzený na 4096.
- Počet MAC adries uložených v prepínacích tabuľkách je obmedzený.
- Riešenie problémov so sieťou bez IP adries a ICMP môže byť ťažké.
- Konfigurácia a prevádzka ARP môže byť na veľkých L2 sieťach komplikovaná.

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
Financované Európskou úniou NextGenerationEU

- Migrácia MAC adries (migrácia inštancie) na iné fyzické lokality predstavuje potenciálny problém, ak správne nenastavíte časové limity tabuľky ARP.

L2 vrstva disponuje obmedzenými nástrojmi na správu siete. Je náročné riadiť prenos, pretože nemá mechanizmy na správu siete alebo formovanie prenosu. Riešenie problémov so sieťou je tiež nepríjemné, čiastočne preto, že sieťové zariadenia nemajú žiadne IP adresy. V podstate neexistuje žiadny rozumný spôsob, ako skontrolovať oneskorenie v sieti [4].

Benefity L3 vrstvy

Výhody použitia L3 sieťovej architektúry sú nasledujúce:

- L3 siete poskytujú rovnakú úroveň odolnosti a škálovateľnosti ako internet
- Ovládanie sieťového prenosu pomocou smerovacích metrík je jednoduché
- Možnosť nakonfigurovať BGP konfederáciu kvôli škálovatelnosti. Vďaka tomu budú mať stav úmerný počtu rackov, nie počtu serverov alebo inštancií.
- Existencia celého radu dobre otestovaných nástrojov napríklad ICMP protokol na monitorovanie a správu prenosu.
- Architektúry na L3 vrstve umožňujú použitie QoS na správu výkonu siete [4].

Limitácie L3 vrstvy

Hlavné obmedzenie sietí L3 vrstvy spočíva v tom, že neexistuje žiadny zabudovaný izolačný mechanizmus porovnateľný s VLAN v L2 vrstve. Hierarchická povaha IP adries znamená, že inštancia je v rovnakej podsieti ako jej fyzický host, čo sťažuje migráciu. Z týchto dôvodov musí sieťová virtualizácia používať IP zapuzdrovanie a softvérové riešenie na koncových hostoch, ktoré slúži na izoláciu a oddelenie adresovania vo virtuálnej vrstve of adresovania vo fyzickej vrstve. Medzi ďalšie potencionálne nevýhody sieťového prepojenia L3 vrstvy patrí potreba navrhnúť schému IP adresovania namiesto spoliehania sa na to, že prepínače budú automaticky sledovať MAC adresy, a konfigurácia smerovacieho protokolu vnútornej braný v prepínačoch [4].

2.4. Typy sieti

Vybraný hardvér servera musí mať vhodný počet a typ sieťových portov, aby podporoval navrhovanú architektúru. Aj keď každá inštalácia OpenStacku je špecifická pre jej použitie, odporúčajú sa nasledujúce siete:

Interná alebo riadiaca sieť – Používa sa ako interná komunikačná sieť medzi výpočtovými a riadiacimi uzlami. Môže sa tiež použiť na komunikáciu iSCSI medzi výpočtovými uzlami a uzlami úložiska iSCSI. Malo by to byť minimálne 1 GB NIC a malo by ísť o nesmerovanú sieť. Toto rozhranie by malo byť redundantné pre vysokú dostupnosť (HA).

Sieť nájomcov (tenant) – súkromná sieť, ktorá umožňuje komunikáciu medzi inštanciami každého nájomcu. Táto sieť by mala byť tiež izolovaná od všetkých ostatných sietí z dôvodu zabezpečenia. Pre HA by malo byť dostatočné redundantné 1Gb rozhranie. Pojem nájomca (tenant) predstavuje v OpenStacku staršie označenie pre projekt v Keystone službe. Od API verzie 3 je projekt preferovaný pojem. Viac o sieti nájomcov a ich typoch v kapitole 6.4.

Úložná sieť – súkromná sieť, ktorú je možné pripojiť k frontendu Ceph alebo inému zdieľanému úložisku. Za účelom HA by malo ísť o redundantnú konfiguráciu s odporúčanými 10Gb sieťami. Táto sieť izoluje úložisko inštancií mimo iných sietí. Pri zaťažení by táto prevádzka úložiska mohla preťažiť iné siete a spôsobiť výpadky v iných službách.

Externá alebo verejná sieť – Táto sieť sa používa na externú komunikáciu z virtuálnych strojov do verejného sieťového priestoru. Tieto adresy zvyčajne spracúva neutrónový agent na riadiacich uzloch, ale môže ich spracovávať aj iný SDN. Ak však

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
Financované Európskou úniou NextGenerationEU

používate neutrónový DVR s OVS, musí byť táto sieť prítomná vo výpočtovom uzle, pretože severnú a južnú prevádzku nebudú spracovávať riadiace uzly, ale samotný výpočtový uzol [4].

2.5. Sieť nájomcov (Tenant network)

Vytvárajú ju používatelia, na účely prepojenia v rámci projektov. Štandardne sú úplne izolované a nezdieľajú sa s inými projektami. OpenStack podporuje niekoľko typov týchto sieti.

- Flat – všetky inštancie sa nachádzajú v rovnakej sieti, ktorú je možné zdieľať s hostom. Nie je tu žiadne VLAN značenie alebo iná segregácia siete.
- VLAN – podpora vytvárať viac sieti pomocou VLAN ID, ktoré zodpovedajú VLAN sieťam prítomným vo fyzickej sieti.
- VXLAN – používajú sieťové prekrytia na podporu privátnej komunikácie medzi inštanciami. Na umožnenie prenosu mimo siete nájomcov a k externým sieťam vrátane internetu je potrebný smerovač. Smerovač poskytuje možnosť pripojiť sa k inštanciám priamo z externej siete pomocou floating IP [4].

2.6. Plugin Modular Layer 2 (ML2)

Plugin modular layer 2 (ml2) je OpenStack Neutron plugin, ktorý umožňuje využívať L2 technológie nachádzajúce sa v komplexných dátových centrách. V súčasnosti podporuje existujúcich L2 agentov ako openvswitch a linuxbridge.

2.7. OpenVSwitch (OVS)

Open vSwitch (OVS) je open-source implementácia prepínača, ktorá beží na Linuxe. Umožňuje vytvoriť viacej prepínačov, VLANky, rozhrania a virtuálne stroje. Jednou z najdôležitejších funkcií OVS je , že je to distribuovaný prepínač. To znamená, že môžeme mať niekoľko virtuálnych strojov na rôznych hypervízoroch a OVS umožní ich vzájomné prepojenie.

Na prepojenie virtuálnych strojov k fyzickej sieti používa OVS tri mosty. Zavádza ďalšie virtuálne rozhrania vnútri hosta identifikované ako mosty a klasifikované nasledovne:

- **br-int**: Predstavuje integračný most. K tomuto mostu sa pripájajú všetky virtuálne inštancie, ktoré používajú sieťovú službu. Patria sem virtuálne inštancie, smerovače a DHCP server.
- **br-tun**: V prípade tunelovaných sietí, br-ethX je nahradený br-tun, ktorý slúži na spracovanie zapuzdrenia a odpuzdrenia paketov.
- **br-ex**: Tento most je pripojený k fyzickému rozhraniu, ktoré poskytuje pripojenie k vonkajšiemu svetu.

Virtuálne stroje sa pripájajú k integračnému mostu pomocou VLAN segmentácie. VLAN id používané na pripojenie VM k integračnému mostu sú lokálne vo výpočtovom uzle. Ak je cieľový VM na rovnakom výpočtovom uzle, integračný most prepne paket lokálne na port cieľového VM. Ak je cieľový port na inom uzle, paket sa prenesie na most br-tun cez patch port. Br-tun zapuzdrí L2 paket vo VXLAN a zamení miestnu VLAN s VXLAN tunel ID, ktorá je alokovaná ML2 VXLAN ovládačom pre virtuálnu sieť. Po dosiahnutí cieľového uzla br-tun dekapsuluje hlavičku, zamení VXLAN ID za lokálne VLAN ID a odovzdá ho br-int, ktorý doručí paket na cieľový port [4].

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
Financované Európskou úniou NextGenerationEU

2.8. Uzly služby OpenStack

Celá koncepcia služby OpenStack sa skladá z riadiaceho uzla (nazývaný controller), výpočtových uzlov (nazývaný compute) a úložiskových uzlov. Aj keď je OpenStack navrhnutý na nasadenie na viaceré fyzické stroje, nie je to vždy potrebné. Systém OpenStack je možné nasadiť iba na jeden server, ktorý bude slúžiť ako spoločný fyzický server pre všetky uzli služby OpenStack.

2.8.1. Riadiaci uzol (Controller node)

Tento uzol hostí komunikačné služby, ktoré podporujú prevádzku celého cloudu vrátane databáz, Dashboardu Horizon a prípadne monitorovacieho systému. Tento uzol môže voliteľne hostiť službu plánovača Nova a zaťaženie serverov. Na zaistenie redundancie musia v klastri existovať najmenej dva uzly radiča [63]

2.8.2. Výpočtový uzol (Compute node)

Tento uzol je hostiteľom hypervízora a virtuálnych inštancií a poskytuje im výpočtové prostriedky. Výpočtový uzol môže slúžiť aj ako sieťový radič pre inštancie, ktoré hostí, ak sa používa sieťová schéma s viacerými hostiteľmi. Môže tiež hosťovať nenáročné interné služby OpenStack [63].

2.9. Vysoká dostupnosť systému OpenStack

Jedným zo spôsobov dosiahnutia vyššej úrovne vysokej dostupnosti je zdvojenie riadiaceho uzla (kontrolér). Týmto krokom dosiahneme odstránenie SPOF tým, že služby riadiaceho uzla budú dostupné aj v prípade, že jeden z riadiacich uzlov prestane fungovať. Tieto dva servery budú prevádzkované ako jeden spoločný klaster, ktorý bude fungovať, podľa možností, v active-active konfigurácii a záťaž serverov bude zdieľaná na oba uzly rovnomerne.

Pri vytváraní kontrolér klastra pre službu OpenStack je potrebné doinštalovať okrem základných programov pre beh služby OpenStack aj ďalšie programy, ktoré slúžia na samotnú klasterizáciu a redundanciu celého systému. Kompletný postup inštalácie a konfigurácie je priložený ako príloha tejto práce. V prílohe je podrobne spísaný postup nasadenia kompletnej OpenStack kontroler-compute topológie spolu s inštaláciou a konfiguráciou podporných programov pre klasterizáciu riadiaceho uzla.

V podkapitolách nižšie je sú popísané programy a technológie, ktoré sú potrebné pre spustenie základného OpenStacku a takisto programy, ktoré je možné použiť na klasterizáciu riadiacich uzlov.

2.9.1. RabbitMQ

RabbitMQ je softvér na sprostredkovanie správ s otvoreným zdrojom (niekedy sa nazýva aj middleware zameraný na správy), ktorý pôvodne implementoval protokol AMQP (Advanced Message Queuing Protocol) a od tej doby bol rozšírený o doplnkovú architektúru na podporu protokolu STOMP (Streaming Text Oriented Messaging Protocol), Message Queuing Telemetry Transport (MQTT) a ďalšie protokoly.

RabbitMQ je napísaný v programovacom jazyku Erlang a je postavený na platforme Open Telecom Platform pre klastrovanie a prepnutie po zlyhaní. Knižnice klientov sú k dispozícii pre všetky hlavné programovacie jazyky.

2.9.2. MariaDB – Galera klaster

MariaDB Galera klaster je pre MariaDB prakticky synchrónny multi-master klaster. Je k dispozícii iba v systéme Linux a podporuje iba úložný modul InnoDB . OpenStack využíva databázovú službu MariaDB v podstate vo všetkých svojich súčastiach a každý modul inštalovaný na riadiaci uzol bude mať svoju vlastnú databázu. MariaDB je inštalovaná na všetky riadiace uzly. Vlastnosti:

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
Financované Európskou úniou NextGenerationEU

- Synchrónna replikácia
- Aktívna-aktívna multi-master topológia
- Čítajte a zapisujte do ľubovoľného uzla klastra
- Automatické riadenie členstva, zlyhanie uzlov vypadávajúcich z klastra
- Automatické pripájanie uzlov
- Paralelná replikácia na úrovni riadkov
- Priame pripojenie klientov, natívny vzhľad z MariaDB

2.9.3. Keepalived

Klaster s vysokou dostupnosťou sa zvyčajne skladá z dvoch serverov: aktívneho primárneho servera a pohotovostného sekundárneho servera. Tieto dva servery zdieľajú rovnaké VIP (virtuálne adresy IP), ktoré sú platné iba pre primárny server. Keď zlyhá primárny server, sekundárny server prevezme VIP, aby mohol naďalej poskytovať služby. Tento režim je široko používaný v prepínači zdrojov a replík MySQL a v prístupe na web Ngnix.

Keepalived je softvér s vysokou dostupnosťou na báze VRRP(Virtual Router Redundancy Protocol), ktorý sa dá použiť na vytvorenie klastra s vysokou dostupnosťou. V tradičných fyzických sieťach je možné primárny / sekundárny stav dohodnúť s protokolom VRRP od Keepalived. Primárne zariadenie periodicky zasiela správy ARP na vyčistenie tabuľky MAC alebo tabuľky terminálovej ARP uplinkovej výmeny, aby sa spustila migrácia VIP na primárne zariadenie [64].

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
Financované Európskou úniou NextGenerationEU

3. AUTOMATIZOVANÉ PRODUKČNÉ NASADENIE PLATFORMY OPENSTACK

Prostredie sa stáva produkčným dňom nasadenia do živej prevádzky. Vo väčšine prípadov dňom prechodu z testovacieho prostredia. Ak by nebol jasný vyššie uvedený pojem „živá prevádzka“, ide o stav, kedy nami alebo inou spoločnosťou vyvíjané prostredie po jeho otestovaní nasadíme a buď mu je pridelená nejaká funkcia, ktorú vykonáva, alebo je poskytnutý priamo zákazníkom, zamestnancom na používanie alebo mu je pridelený iný účel, častokrát komerčný. V našom prípade je takéto prostredie práve to cloudové, konkrétnejšie OpenStack. Cloudové prostredie v distribúcii OpenStack, ako už bolo spomenuté, ponúka 2 prístupy, účely pre jeho vytvorenie. V cloudovom svete známe ako: Private cloud (privátny cloud) a Public cloud (verejný cloud).

Produkčné prostredie v prvom prípade, a to pri vytvorení privátneho cloudu, je väčšinou používané na súkromné účely, čo vyplýva aj z jeho názvu private (privátny, súkromný). Využitie privátnych cloudov je pestré a väčšinou za jeho poskytovanie neplatíme a ani nežiadame žiadne finančné príspevky od klientov. Firmy a podniky takéto prostredia využívajú pre vnútorné systémy, infraštruktúru, výučbu a podobne.

Na druhú stranu public cloud, alebo verejný cloud, slúži, ako už z jeho názvu vyplýva, verejnosti. Vo väčšine prípadov je takéto prostredie vytvorené za komerčným účelom, poskytovať firmám a podnikom celé prostredie alebo len jeho zdroje. Veľmi dobrým príkladom takéhoto prostredia, ako už bolo spomenuté, je Amazon Web Services. AWS svoje prostredie poskytuje verejnosti za určité spoplatnenie, z čoho zákazník profituje, pretože sa nemusí starať o dostupnosť všetkých prostriedkov. Napríklad firma xy chce vytvoriť alebo premigrovať celú sieťovú infraštruktúru do cloudu. Ak by mala vytvoriť privátny cloud a na ňom potom prevádzkovať alebo doňho zmigrovať sieťovú infraštruktúru, bolo by to veľmi časovo a finančne náročné, pretože by museli vybudovať 2 prostredia nanovo. A toto je presne chvíľa, v ktorej využitie verejného cloudu vyniká. Poskytovateľ verejného cloudu firme poskytne prostredie (virtuálne servery atď.) za určitý finančný, vo väčšine prípadov hodinový, poplatok a firma môže bezstarostne začať s migráciou. No v prípade, že my ako firma sme tvorcom a poskytovateľom verejného cloudu, tak aj toto je jedna z našich úloh. Poskytovať zdroje, cloudové prostredie verejnosti.

OpenStack v produkčnom nasadení sa stáva populárnejší vďaka jeho vysokej podpore, údržbe a tomu, že sa jedná o free open-source projekt. Jedno z jeho prvých produkčným prostredí vôbec vzniklo v spoločnosti, ktorá sa priamo podieľala aj na jeho vývoji, a tou je NASA [18]. Jej úsilie štandardizovať svoje webové stránky inšpirovalo prelom v cloud computing technológii. Nakoniec sa z dôvodu komplexnosti ich úlohy dostali až k téme cloud computingu a vznikol projekt Nebula. Práve členovania projektu Nebula stoja za tvorením cloud computing prostredia nazývaným Openstack.

V dnešnej dobe už značná časť prostriedkov spoločnosti NASA beží v cloudovom prostredí AWS, no OpenStack stále využívajú v NASA Center for Climate Simulations (NCCS). Na jeho nasadenie a management používajú Ansible. Takéto prostredie pozostáva z 9000 cores a 322 teraflops výpočtového výkonu [19].

Prekvapením môže byť aj to, že mimo NASA používajú produkčné nasadenie OpenStack aj iné vládne organizácie ako štátna služba Spojených štátov Amerických, alebo francúzske ministerstvo vnútra.

Známych firiem, ktoré už dnes prevádzkujú alebo využívajú OpenStack cloud je nespočetne veľa. Medzi také známe zahraničné spoločnosti patria: Cert, Wallmart, NASA, Yahoo, Target, T-mobile, Nike a mnoho ďalších [19].

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
Financované Európskou úniou NextGenerationEU

Medzi česko-slovenské firmy, ktoré prevádzkuje OpenStack cloud patrí aj seznam.cz [20]. OpenStack prostredie im beží v troch dátových centrách, ktoré pozostávajú z 12tisíc fyzických serverov.

3.1. Nástroje na automatizované nasadenie OpenStacku

Inštalácia Openstack-u je odlišná od inštalácie nejaké programu do zariadenia, je to oveľa komplexnejšia záležitosť a preto sa pri inštalácií využívajú nástroje a tzv. frameworky, ktoré pomáhajú udržiavať životný cyklus nasadeného OpenStack-u. Framework je štruktúra, ktorá slúži ako základ pre budovanie softvéru a jej výhodou je, že sa nezačína úplne od nuly (from scratch). V súčasnosti existuje niekoľko frameworkov [21]:

TripleO

TripleO je názov pre „OpenStack na Openstack-u“. Ide o oficiálny projekt OpenStack s cieľom umožniť nasadenie a spravovanie produkčného cloudu na fyzickom hardvéri pomocou existujúcich komponentov OpenStack-u. Pri TripleO sa začne s vytvorením „undercloud“, ktorý bude obsahovať potrebné komponenty OpenStack-u na nasadenie a správu „overcloud“ respektíve pracovného cloudu. Overcloud je nasadené riešenie a môže predstavovať cloud na akýkoľvek účel. TripleO využíva niekoľko existujúcich základných komponentov OpenStack ako: Nova, Ironic, Neutron, Heat, Glance a Ceilometer.

OpenStack-Helm

OpenStack-Helm je riešenie nasadenia OpenStack-u pomocou Helm-u s využitím Kubernetových kontajnerov. Helm je nástroj, ktorý sa využíva na definíciu, inštaláciu a inovovanie aplikácií bežiacich na Kubernetoch. Vo svojej najzakladanejšej podobe je Helm nástroj na vytváranie šablón. Kubernetes sa využíva na automatizácie prevádzkových úloh správy kontajnerov a obsahuje vstavané príkazy na nasadzovanie aplikácií, zavádzanie zmien v aplikáciách a škálovanie aplikácií, či už smerom nahor alebo nadol. Na nasadenie Kubernetového klastra pre produkčné prostredie sa odporúča použiť vysoko odolnú distribúcia ako Airship alebo KubeADM.

Kolla-Anisble

Kolla-Ansible nasadzuje kontajnerovú riadiacu rovinu OpenStack pomocou kontajnerov Kolla, organizovaných cez Ansible. Projekt sa zameriava na jednoduchosť a spoľahlivosť a zároveň poskytuje flexibilný a intuitívny model konfigurácie. Kolla poskytuje kontajnery a nástroje na prevádzku cloudov OpenStack, ktoré sú škálovateľné, rýchle, spoľahlivé a inovovateľné pomocou osvedčených postupov komunity. Ansible je softvérový nástroj, ktorý poskytuje jednoduchú, ale výkonnú automatizáciu pre multiplatformovú počítačovú podporu.

Bifrost

Bifrost je súbor príručiek Ansible, ktorý pomocou komponentu Ironic automatizujeúlohu nasadenia základného obrazu na hardvér. Poskytuje modulárny nástroj pre jednorazové nasadenie operačného systému s čo najmenším počtom prevádzkových požiadaviek.

Kayobe

Kayobe nasadí kontajnerovú riadiacu rovinu OpenStack na fyzické zariadenia. Bifrost sa používa na objavovanie a poskytovanie cloudových serverov. Kolla sa používa na vytváranie obrazov kontajnerov pre služby OpenStack. Kolla Ansible sa používa na nasadenie kontajnerovej riadiacej roviny OpenStack.

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
Financované Európskou úniou NextGenerationEU

Openstack-Ansible

OpenStack-Ansible poskytuje Ansible playbooky a roly pre nasadenie a konfiguráciu prostredia OpenStack. Teda oproti nasadenie Kolla-Ansible nebudeme využívať kontajnery, ale len nástroj Ansible a komponenty OpenStack-u.

OpenStack-Charms

Openstack-Charms je kolekcia Charmed Operators, nazývaných tiež jednoducho „charms“, ktoré sa používajú na nasadenie a správu cloudov OpenStack pomocou MAAS a Juju. Každý OpenStack charm je zodpovedný za nasadenie a správu životného cyklu jedinej cloudovej služby (napríklad charm nova-compute pre službu Nova Compute). Projekt zahŕňa aj charmy pre vybraný softvér, ktoré nepodporuje OpenStack, ako sú Ceph, MySQL a RabbitMQ.

OpenStack-Chef

Chef je platforma pre správu systémov s otvoreným zdrojom a automatizáciu cloudovej infraštruktúry. Nasadenie OpenStack-u pomocou Chef-u by malo obsahovať „cookbooks“.

3.2. OpenStack Charms

Pre samotnú inštaláciu OpenStack-u bola zvolená ako najvhodnejšia metóda OpenStack-Charms, Charmed Operators alebo tiež nazývane „Charms“ sa používajú na nasadenie a správu cloudov OpenStack pomocou Metal-as-a-Service (MAAS) a Juju. OpenStack Charms vzniklo vďaka spoločnosti Canonical, ktorá stojí za vývojom operačného Linux systému Ubuntu, a taktiež sa priamo podieľa na vývoji samotného OpenStack projektu, takže toto riešenie nasadenia je podporované a odporúčané aj zo strany samotného OpenStacku. Aj na oficiálnej stránke OpenStack je väčšina návodov priamo budovaná a vysvetlená pre tento typ nasadenia, vďaka čomu je pomerne jednoduché nájsť riešenie na vzniknuté problémy. Aj zo strany komunity je tento typ nasadenia odporúčaný a v prípade otázok sú všetci ochotní pomôcť, či už na fórach a stránkach priamo OpenStack projektu, alebo pomoc, návody a dokumentáciu jednotlivých častí (MAAS [21],[24] a Juju [25]) nájdeme na fórach a stránkach Canonical Ubuntu. OpenStack Charms pomenúva jednotlivé moduly ako Charms alebo v preklade „kúzla“. Tieto kúzla sú uložené v CharmHube, ktorý si môžeme prestaviť ako internetový obchod iba s tým rozdielom, že je tam všetko zadarmo. Na CharmHube [22] nájdeme všetky moduly OpenStacku a aj množstvo ďalších aplikácii, ktoré nie sú implementované priamo v OpenStack projekte, ako napríklad Nginx, Grafana, Prometheus, Dokuwiki a mnoho ďalších. Vďaka jednoduchému prístupu, ktorý OpenStack Charms ponúka sme schopní tieto ďalšie aplikácie a služby nasadiť do bežiaceho prostredia či už pomocou jedného komplexného príkazu alebo pomocou Juju webového rozhrania.

OpenStack Charms so všetkým čo prináša zjednodušuje celý proces nasadenia celého cloudového prostredia v distribúcií OpenStack, no taktiež sa nezaobíde bez vedomostí a analýzy možností jeho fungovania

3.2.1. Metal-as-a-Service(MAAS)

Metal-as-a-Service (MAAS) poskytuje kompletnú automatizáciu fyzických serverov pre efektívnosť prevádzky serverov/dátového centra. Softvér je open-source a je zastrešovaný firmou Canonical, ktorá stojí za distribúciou Linux Ubuntu. MAAS zaobchádza s fyzickými servermi ako s virtuálnymi strojmi (inštanciami) v cloude. Toto riešenie umožňuje spravovať niekoľko serverov naraz a z jedného miesta, aby sa nemuseli spravovať jednotlivo a komplikovane. Na diaľku je teda možné nasadenie štandardných alebo upravených operačných systémov na fyzické alebo virtuálne stroje. MAAS komplexne spĺňa potrebu rýchleho nasadenia, zničenia a rekonfigurácie operačného systému.

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
Financované Európskou úniou NextGenerationEU

V súčasnosti je nasadzovaný v bankovom, telekomunikačnom a priemyselnom prostredí, ako aj na špecializované účely, ako je overovanie front-end superpočítačov, streamingové hudobné služby, obnova po havárii a analýza rizík počítačovej bezpečnosti. Hlavnou kľúčovou funkciou serveru MAAS pre naše účely bude automatizácia, kedy pomocou Intelligent Platform Management Interface (IPMI) si MAAS zapne/reštartuje daný stroj a pomocou Preboot eXecution Environment (PXE) buď cez sieť IPv4 alebo IPv6 nainštaluje požadovaný alebo predvolený operačný systém na tento úkon. Ďalšími užitočnými funkciami je konfigurácia úložiska, sieťových rozhraní a manažment IP adries pomocou Dynamic Host Configuration Protocol (DHCP) [24].

3.3. Komponent Juju

Juju je open-source framework, ktorý využíva Charmed operátorov alebo „Charms“ na nasadenie cloudovej infraštruktúry, aplikácií a správu ich operácií od spustenia. S Juju je možné inštalovať, udržiavať, upgradovať a integrovať aplikácie naprieč klastrami, kontajnermi Kubernetes, virtuálnymi a fyzickými strojmi na verejných alebo súkromných cloudoch. Výrobcovia opisujú Juju ako softvér, ktorý riadi váš softvér. Pomáha vám prevziať kontrolu nad všetkými vašimi aplikáciami, infraštruktúrou a prostrediami. Použitie tohto softvéru by malo zabezpečiť:

- ušetrenie času stráveného nad správou skriptov
- minimalizovať náklady
- zabezpečiť redundanciu a odolnosť
- monitorovať všetky aktivity naprieč infraštruktúrou
- maximalizovať architektúru hybridného cloudu

Charmy sú malé aplikácie, ktoré obsahujú bežné sady inštrukcií funkcií, aby zabezpečili opakovateľný a spoľahlivý kód. Toto umožňuje spravovať aplikácie a scenáre namiesto konfigurácií, avšak v prípade potreby je tu stále možnosť ponoriť sa do konfiguračných súborov YAML (YAML Ain’t Markup Language). Charmed Operators sa nevyužívajú len na nasadzovanie a spravovanie jednotlivých aplikácií, ale aj na ich spájanie medzi sebou v modeloch. Aplikačný model definuje, ktoré aplikácie poskytujú službu a ako spolu súvisia. Riadenie Juju je možné pomocou príkazové riadku (CLI) alebo pomocou grafického webového rozhrania (GUI) ako je Juju GUI [25].

Ďalšie informácie ku automatizovanému nasadeniu OpenStack platformy pomocou Charms sú k dispozícií v podpornom dokumente s názvom „Automatizované nasadenie OpenStack“.

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
Financované Európskou úniou NextGenerationEU

4. PLATFORMA PRE SPRÁVU CLOUDU

Spravovanie cloudov môže byť náročné. Softvér platforiem pre správu cloudu je navrhnutý tak, aby túto náročnosť znížil. Zvyčajne poskytuje nasledujúce funkcie:

- Prezeranie a manažovanie viacerých cloudov v rovnakom prostredí.
- Vytváranie a spravovanie virtuálnych strojov v integrovanom prostredí. Je schopný poskytovať šablóny na rýchle vytvorenie virtuálnych strojov.
- Ponúka služby naprieč viacerými cloudmi
- Má schopnosť využívať existujúcu IT infraštruktúru.
- Poskytuje jednotné správy a analýzy týkajúce sa využitia cloudov.
- Poskytuje nasadenie aplikácií naprieč viacerými prostrediami

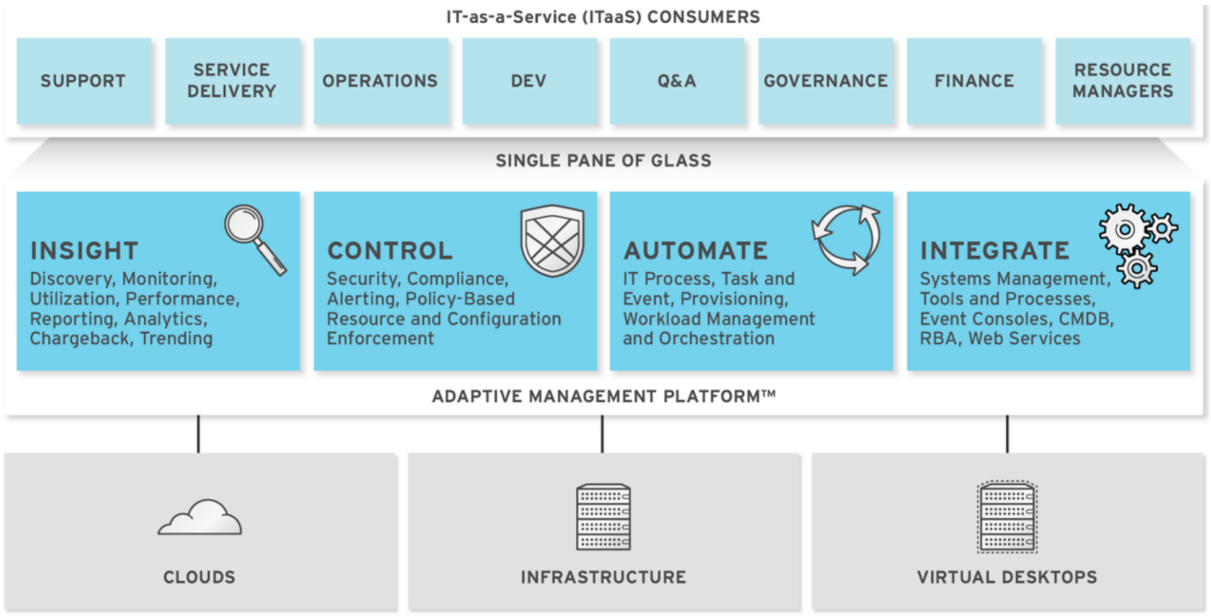
Výber platformy závisí od funkcií poskytovaných softvérom a skúsenosti s používaním. Niektoré sú určené pre konkrétneho hypervízora a iné môžu byť použiteľné s rôznymi hypervízormi. Niektoré z nich sa používajú ľahko, iné môžu vyžadovať viac praktických skúseností. Výber platformy závisí od typov úloh, ktoré je potrebné splniť, od rozpočtových obmedzení, podpory dodávateľov softvéru a kompatibility s cloudovými technológiami používanými v hybridnom cloude [2].

4.1. ManageIQ

ManageIQ je open-source platforma pre manažovanie hybridnej cloudovej infraštruktúry. Môže byť nasadená na správu malých ako aj veľkých infraštruktúr s podporou technológií ako sú virtuálne stroje, privátne/verejné cloudy a kontajnery. Nasadenie ManageIQ je umožnené viacerými spôsobmi a to ako Vagrant box, Docker kontajner, Kubernetes cluster alebo vo verejnom cloude. Inštalácia každým spôsobom je relatívne jednoduchá, pre POC riešenie sme zvolili Docker kontajner [66][65]. Našim motívom nie je vyskúšať rôzne spôsoby nasadenia, ale jednoducho pochopiť architektúru a funkcionalitu, ktorá je užitočná pre náš hybridný cloud.

ManageIQ alebo iné podobné orchestračné platformy komunikujú s cloudmi pomocou API kľúčov. Tie majú rôzne formy v závislosti od poskytovateľa cloudu. Napríklad pri AWS je to kombinácia access key a secret key. Pre GCP je to špeciálny JSON súbor. OpenStack používa meno a heslo používateľa zadefinované v službe keystone. Nech už to je čokoľvek, je to jednoducho spôsob autentifikácie API volaní uskutočnených ManageIQ platformou.

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
Financované Európskou úniou NextGenerationEU



Obrázok 4.1 Prehľad vlastností a funkcií ManageIQ [5]

ManageIQ poskytuje integráciu s nasledujúcimi cloudovými poskytovateľmi: Amazon Web Services, Microsoft Azure, Google Cloud Platform, RedHat Openstack Platform, IBM PowerVC, IBM Power Systems Virtual Servers, IBM Cloud VPC, Oracle Cloud.

Taktiež ponúka integráciu s kontajnerzačnými poskytovateľmi: OpenShift Container Platform, Kubernetes a Amazon Elastic Kubernetes Service.

V neposlednom rade dokáže integrovať aj s technológiami na virtualizáciu infraštruktúry: VMware v Sphere, Red Hat Virtualization, Microsoft System Center Virtual Machine Manager, OpenStack Platform Director.

4.1.1. Architektúra ManageIQ

Architektúra platformy pozostáva z niekoľkých komponentov:

- Aplikácia ManageIQ, ktorá je nasadená vo forme bezpečnej, vysoko-výkonnej, predkonfigurovanej virtuálnej mašiny s natívnou podporou pre HTTPS komunikáciu.
- Server ManageIQ, ktorý je súčasťou aplikácie. Je to softvérová vrstva, ktorá zabezpečuje komunikáciu medzi SmartProxy a virtuálnou databázou správy systému (Virtual Management Database - VMDB). Taktiež podporuje HTTPS komunikáciu.
- Virtuálna databáza správy systému, ktorá je buď priamo súčasťou aplikácie, prípadne na inom serveri, ktorý je dostupný pre aplikáciu. Dáta v databáze predstavujú informácie o celkovej virtuálnej infraštruktúre integrovanej s ManageIQ. Takisto udržuje informácie o aktuálnom stave úloh, ktoré v aplikácii prebiehajú.
- Konzola ManageIQ je webové rozhranie, ktoré sa používa na prehľad a správu ManageIQ Servera a aplikácie.
- SmartProxy môže byť nasadená priamo na aplikácii alebo na ESX serveri. V prípade, že nie je súčasťou ManageIQ servera, môže byť nasadená priamo z aplikácie. SmartProxy Agent musí byť nakonfigurovaný na každej úložiskovej lokácii a musí byť viditeľný pre aplikáciu. Tento komponent komunikuje s aplikáciou prostredníctvom HTTPS.

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
Financované Európskou úniou NextGenerationEU

4.1.2. Požiadavky na platformu

Na používanie ManageIQ platformy je nutné splniť niekoľko určitých požiadaviek na hardvér, databázu a prehliadač.

Minimálne hardvérové požiadavky:

- 4 vCPUs
- 12 GB RAM
- 44 GB HDD + voliteľná veľkosť

Databázové požiadavky

RedHat odporúča alokovať disk pre virtuálnu mašinu perzistentne v čase jej tvorby. Na veľkosť databázy vplývajú hlavne tri faktory:

- Počet virtuálnych mašín – najdôležitejší faktor, ktorý ma vplyv na veľkosť databázy (VMDB).
- Počet hostiteľských serverov (Host count).
- Počet individuálnych úložiskových serverov (nie je to celkový počet virtuálnych diskov patriacich pod virtuálne mašiny).

Na odhad veľkosti potrebnej pre databázu sa môže použiť táto tabuľka [5]:

Tabuľka 4.1 Tabuľka na odhadnutie potrebnej veľkosti disku pre databázu ManageIQ

Počet VM	Počet serverov	Počet úložiskových serverov	Po 1 roku	Po 2 rokoch
100	5	50	3.5 GB	5 GB
500	10	100	17 GB	25 GB
5000	50	500	173 GB	251 GB

Požiadavky na prehliadač

ManageIQ podporuje na prístup ku svojmu grafickému webovému rozhraniu bežné webové prehliadače ako Google Chrome, Mozilla Firefox, Safari, Internet Explorer 10+.

Možnosti nasadenia

V produkčných prostrediach je platforma prevažne nasadzovaná stiahnutím predpripravenej virtuálnej mašiny a následne jej nasadením do virtualizačnej platformy ako je OpenStack alebo VMware. Nasleduje jej pripojenie ku virtualizačným platformám, aby mohli byť spravované pomocou ManageIQ.

Okrem už tradičnej formy virtuálnych mašín, ManageIQ poskytuje možnosť nasadenie vo forme kontajnerov bežiacich buď v Kubernetes alebo OpenShift. Nižšie je uvedený zoznam platforiem kompatibilných s ManageIQ.

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
Financované Európskou úniou NextGenerationEU

Tabuľka 4.2 Zoznam podporovaných platforiem ManageIQ

Platforma	Virtuálna mašina / Pod
Amazon Web Services (AWS)	VM
Microsoft Azure	VM
Google Cloud Platform (GCP)	VM
Microsoft System Center Virtual Machine Manager (SCVMM)	VM
Kubernetes	Pod
Openshift Container Platform (OCP)	Pod
Red Hat OpenStack Platform (OSP)	VM
Red Hat Virtualization (RHV)	VM
VMware vSphere	VM

Ďalším spôsobom nasadenia je využitie docker kontajnerov s ľubovoľným nasadením. Po nasadení aplikácie je možné sa do nej prihlásiť a spravovať ju. Aplikácia pri jednoduchej inštalácii ponúka webové grafické rozhranie, ktoré beží na IP adrese, prípadne doménovom mene virtuálnej mašiny, na porte s číslom 8443.

4.2. Centralizovaná autentifikácia ManageIQ

Centralizovaná autentifikácia predstavuje spôsob riadenia prístupu používateľov k informačným systémom prostredníctvom jedného zdroja overenia identity. V tomto modeli sa používateľ autentifikuje len raz voči centrálnemu autentifikačnému serveru, ktorý následne umožňuje prístup k viacerým službám. Kľúčovým prvkom tohto systému je **LDAP (Lightweight Directory Access Protocol)**, ktorý slúži na prístup k adresárovým službám obsahujúcim údaje o používateľoch, ich roliach, skupinách a oprávneniach. LDAP umožňuje efektívne vyhľadávanie a overovanie prihlasovacích údajov v hierarchickej štruktúre záznamov, čím zabezpečuje jednotnú správu identít a výrazne zjednodušuje administráciu v rozsiahlych sieťach. Tento protokol je široko podporovaný a často využívaný v kombinácii s ďalšími bezpečnostnými mechanizmami, ako napríklad SSL/TLS šifrovaním, pre zvýšenie ochrany autentifikačných údajov.

4.2.1. Konfigurácia autentifikácie v prostredí ManageIQ

V testovacej inštancii bola použitá verzia ManageIQ *Petrosian-1*, prevádzkovaná na operačnom systéme CentOS Stream 8 v rámci virtualizačnej platformy VMware. Inštancia je nakonfigurovaná s dvoma virtuálnymi diskami – osobitne pre systém a pre používateľské dáta. LDAP/AD autentifikácia je realizovaná prostredníctvom externého AD servera. Z dôvodu absencie dôveryhodného certifikačného úradu bolo v konfiguračnom súbore služby SSSD použité nastavenie ldap_tls_reqcert = never, čím sa deaktivovalo overovanie TLS certifikátu. Pre zabezpečenie funkcionality boli na sieťovej vrstve otvorené porty 389 (LDAP) a 636 (LDAPS).

Zhrnutie komponentov a parametrov:

- ManageIQ verzia: *Petrosian-1*
- OS: CentOS Stream 8
- Virtualizácia: VMware
- LDAP porty: 389 (LDAP), 636 (LDAPS)
- TLS certifikát: deaktivované overovanie (ldap_tls_reqcert = never)

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
Financované Európskou úniou NextGenerationEU

4.2.2. Inštalácia a konfigurácia SSSD

Na konfiguráciu autentifikácie bol použitý balík SSSD s doplnkovými nástrojmi (realmd, oddjob, krb5-workstation, a pod.). Konfigurácia vychádzala z oficiálnej dokumentácie ManageIQ a manuálových stránok nástrojov sssd-ldap.conf, ldap.conf a authconfig.

Po inštalácii balíkov sa overila dostupnosť LDAP servera prostredníctvom nástroja ldapsearch. Komunikačný test prebiehal cez LDAPS protokol s deaktivovaným overovaním certifikátov.

Na základe šablóny poskytovanej ManageIQ bol vytvorený konfiguračný súbor sssd.conf. V rámci konfiguračných parametrov boli definované:

- LDAP URI,
- spôsob overovania a mapovania identít,
- viazanie na konkrétnu doménu a atribúty objektov používateľov a skupín,
- správa Kerberos realm-u.

Doménové parametre špecifikovali použitie schémy Active Directory a atribúty ako sAMAccountName, displayName, mail či msSFU30UidNumber.

Po úpravách bol systém reštartovaný (systemctl restart sssd) na načítanie nových nastavení.

4.2.3. Testovanie konfigurácie

Pre overenie správnej funkčnosti konfigurácie boli využité nástroje ako dbus-send, id, getent passwd a sss_cache. Príkazy umožnili overiť prístup k používateľským a skupinovým atribútom, ako aj funkčnosť vyhľadávania prostredníctvom NSS (Name Service Switch). Dočasné deaktivovanie SELinux kontrol (setenforce 0) umožnilo vykonať D-Bus dotazy v testovacom režime.

Pri analýze chýb boli kľúčové log súbory:

- /var/log/sssd/sssd.log
- /var/log/sssd/sssd_*.log
- /var/log/sssd/sssd_ifp.log

V prípade potreby bolo možné zvýšiť úroveň ladeni prostredníctvom parametra debug_level.

4.2.4. Integrácia s Apache a ManageIQ UI

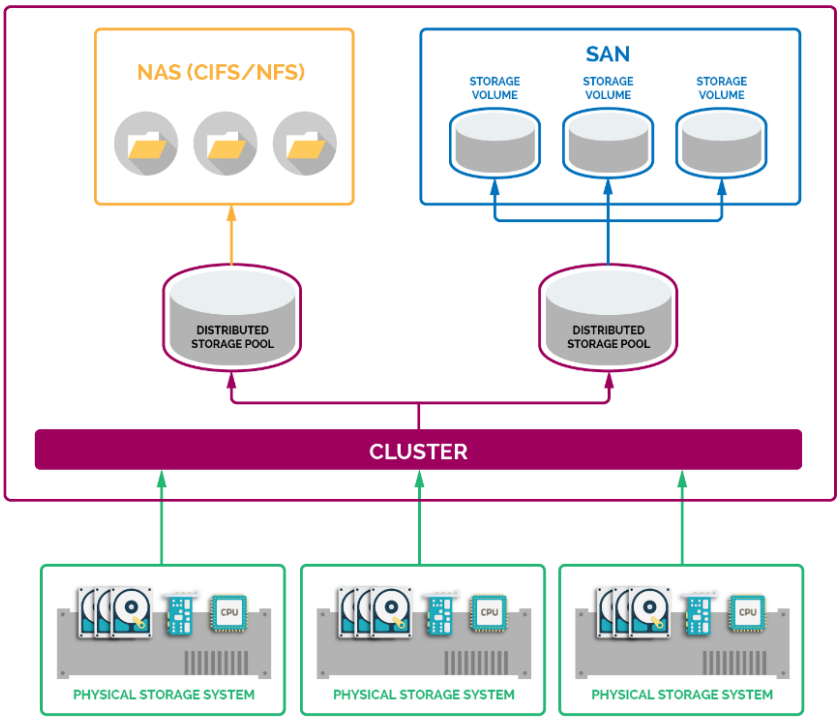
Súčasťou integrácie autentifikácie bolo nasadenie konfiguračných súborov pre Apache server a úprava pravidiel SELinux. V prostredí ManageIQ sa vykonali nastavenia skupín a prístupových práv v rozhraní *Access Control*, kde sa zabezpečila synchronizácia LDAP skupín a ManageIQ rolí. Autentifikačný režim bol zmenený na *External (httpd)* s povoleným načítaním skupín zo servera.

Po úspešnej konfigurácii a testovaní bolo prostredie pripravené na zabezpečený prístup používateľov prostredníctvom centrálnej LDAP/AD autentifikácie. Táto implementácia predstavuje efektívny a škálovateľný spôsob riadenia prístupu v rámci manažovateľného cloudového prostredia.

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
Financované Európskou úniou NextGenerationEU

5. DISTRIBUOVANÉ ÚLOŽISKO

Tento prístup k ukladaniu údajov sa vyznačuje decentralizáciou, škálovateľnosťou a spoľahlivosťou a ponúka komplexné riešenie výziev, ktoré prináša exponenciálny rast digitálnych informácií.



Obrázok 5.1 Všeobecná schéma systému distribuovaného úložiska

5.1. Princípy a výhody distribuovaného úložiska

Distribuované úložisko zabezpečuje decentralizáciu rozložením dát na viacero fyzických serverov, ktoré sa často nachádzajú na rôznych geografických miestach. Tieto servery sú prepojené cez sieť. Táto metóda je v protiklade s tradičnými modelmi ukladania, ktoré centralizujú dáta na jednom mieste. Distribuovaný model zvyšuje rýchlosť prístupu k dátam, spoľahlivosť, škálovateľnosť a odolnosť voči chybám.

Na obrázku Obrázok 5.*1* vyššie je zobrazená základná schéma distribuovaného úložiska, kde je decentralizácia znázornená vo forme viacerých fyzických serverov („physical storage system“) zjednotených do tzv. klastra („cluster“). Prístup do distribuovaného úložiska je možné zabezpečiť pripojením do klastra vo forme súborového sieťového úložiska NAS (Network Attached Storage), alebo blokového sieťového úložiska SAN (Storage Area Network) [27][28][29].

Základom distribuovaného úložiska sú jeho schopnosti bezproblémového rozširovania kapacity (škálovateľnosti), rozdeľovania a replikovania údajov na viaceré servery. Táto konfigurácia umožňuje efektívnu správu dát, paralelný prístup k dátam a ich spracovaniu, čo výrazne znižuje bottleneck-y (úzke hrdlá). Rozdelenie dát alebo „sharding“ je rozdelenie dát na menšie segmenty, ktoré sú distribuované do rôznych serverov úložiska. Okrem toho stratégie replikácie zabezpečujú, že sa v rôznych serveroch udržiava viac kópií uložených dát, čo poskytuje robustnú vrstvu ochrany a dostupnosti údajov, čím sa zabezpečuje spoľahlivosť. Táto replikácia zaručuje, že systém má prístup k dátam z iných nepoškodených serverov aj v prípade zlyhania jedného alebo viacerých serverov. Faktor replikácie môžu nakonfigurovať

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
Financované Európskou úniou NextGenerationEU

administrátori, aby určili počet kópií dát, ktoré systém udržiava. Vyššie faktory replikácie zvyšujú bezpečnosť dát, ale vyžadujú aj viac úložného priestoru [29][30].

5.2. Využitie distribuovaného úložiska

Distribuované úložisko má široké uplatnenie v rôznych odvetviach vrátane médií, zdravotníctva a analýzy veľkých objemov dát. Umožňuje efektívnu manipuláciu s rozsiahlymi knižnicami multimediálneho obsahu, bezpečné ukladanie citlivých údajov pacientov a správu veľkých súborov dát na analytické spracovanie [29][31].

Cloudové platformy vo veľkej miere využívajú distribuované úložisko vďaka jeho škálovateľnosti a odolnosti voči chybám. Amazon S3 je najlepším príkladom distribuovaného úložiska v praxi, ktoré ponúka používateľom vysoko dostupné a bezpečné riešenia úložísk. Distribuované úložisko je obzvlášť užitočné pri riešení rozsiahlych potrieb ukladania a spracovania dát a podporuje širokú škálu aplikácií. S rastúcim objemom dát bude význam distribuovaného ukladania ešte dôležitejší, čo povedie k pokroku a vylepšeniam v technológiách správy dát [31][32].

5.3. Dôvody nasadenia distribuovaného úložiska

Dôvody nasadenia distribuovaného úložiska pre cloud na KIS prakticky kopírujú všetky už vyššie zmienené výhody distribuovaného úložiska. Cloud je nasadený nad platformou OpenStack. Hlavnou výhodou cloudových platforiem je ich všestrannosť, možnosti migrovania virtuálnych zariadení a ich dostupnosť v prípade výpadku zariadenia v klastri. Tieto vlastnosti sú však značne limitované v prípade, že úložisko pre každé dané zariadenie je riešené lokálne. To má za následok, že v prípade prístupu viacerých klientov sa zvyšujú nároky na daný konkrétny server, čím sa okrem pomalšieho spracovania požiadaviek časom zvyšuje riziko zlyhania daného servera. V prípade takého zlyhania vďaka lokálnemu ukladaniu dát, je strata dát a znemožnenie poskytovania služieb, ktoré sa na danom serveri nachádzali, veľmi pravdepodobné.

Nasadením distribuovaného úložiska do cloudu sa týmto riziko straty dát alebo prerušenia poskytovania služieb výrazne znižuje. Práve vlastnosti distribuovaného úložiska zabezpečujú paralelný prístup k dátam a ich výraznú redundanciu. Úložisko už nie je lokálne a vďaka replikácií dát na viaceré serveri nedôjde k strate dát v prípade ich zlyhania. Jednoduchá škálovateľnosť zabezpečuje, že prípadné rozširovanie katedrového klastra je výrazne jednoduchšie. V neposlednom rade sa zjednodušuje aj spravovanie a monitorovanie úložiska. Softvéry distribuovaného úložiska poskytujú nástroje na detailnú správu a monitorovanie, ktoré v takom rozsahu v OpenStack-u nie je možné.

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
Financované Európskou úniou NextGenerationEU

6. VÝBER SOFTVÉRU DISTRIBUOVANÉHO ÚLOŽISKA PRE OPENSTACK

Výber vhodného softvéru pre distribuované úložisko v OpenStack-u je rozhodujúci z niekoľkých dôvodov. Správny backend úložiska je nevyhnutný pre zabezpečenie optimálneho výkonu klastra OpenStack. Ak úložisko nedokáže zabezpečiť potrebné vstupno-výstupné operácie pre vyžadované zaťaženie klastra, vedie to k problémom s výkonom, ktoré priamo ovplyvňujú fungovanie aplikácií. Dobre zvolené riešenie úložiska zvyšuje efektivitu prevádzkovania úložiska, a tým aj poskytovaných služieb. Taktiež môže zjednodušiť procesy údržby, zálohovania a riešenia problémov, zatiaľ čo nesprávne zvolené riešenie môže zvýšiť pracovné zaťaženie administrátorského tímu [33].

Rovnako ako poskytnutie potrebného výpočtového výkonu, zabezpečenie spoľahlivosti riešenia úložiska priamo súvisí so spoľahlivosťou celého klastra. Preto musí byť spoľahlivosť dát najvyššou prioritou pri navrhovaní privátneho cloudu. Výber nesprávneho riešenia úložiska môže tento aspekt ohroziť. Okrem toho ďalším kľúčovým faktorom je nákladová efektívnosť, ktorú treba zohľadniť. Riešenie úložiska by malo vyvažovať atribúty výkonu a nákladov. Plytvanie cennými zdrojmi môže byť dôsledkom riešenia, ktoré zvyšuje celkové náklady bez toho, aby primerane spĺňalo požiadavky na výpočtový výkon určených v príslušných SLA (Service Level Agreement) [33][34].

Ďalším dôležitým aspektom pri výbere softvéru je pochopenie povahy údajov a spôsob, akým je potrebné k nim pristupovať alebo ich distribuovať. Napríklad štruktúrované a neštruktúrované údaje si vyžadujú rôzne riešenia ukladania, čo výrazne ovplyvňuje prístup k údajom a ich ukladanie. Okrem toho je nevyhnutné zvážiť škálovateľnosť a potenciál budúceho rastu riešenia úložiska. Vybrané úložisko by malo nielen vyhovovať súčasným potrebám, ale malo by byť aj škálovateľné, aby vyhovovalo budúcemu rastu požiadaviek na kapacitu a výkon [34].

Výber vhodného softvéru distribuovaného úložiska pre OpenStack si vyžaduje strategický rozhodovací proces, ktorý ovplyvňuje výkon, spoľahlivosť, prevádzkovú a nákladovú efektívnosť celej cloudovej infraštruktúry. Je dôležité zvážiť špecifické potreby a okolnosti každého nasadenia OpenStack-u.

6.1. Kritéria pre výber softvéru

Vypracovanie špecifického súboru kritérií pre výber softvéru je nevyhnutným krokom pre zabezpečenie, aby nami vybraný softvér zodpovedal našim stanoveným cieľom. To nám umožní kvantifikovať a porovnať vlastnosti a schopnosti rôznych softvérov.

Vybraný softvér musí spĺňať nasledujúce kritériá:

- voľne dostupný na použitie
- podpora objektového, blokového i súborového typu úložiska
- škálovateľnosť
- zálohovateľnosť, vhodná redundancia uložených dát
- kompatibilita s platformou OpenStack
- kompatibilita s nástrojmi OpenStack Charms

Tieto kritéria tvoria základný rámec pre výber softvéru s ohľadom na požiadavky, infraštruktúru a architektúru privátneho cloudu poskytovaného na KIS.

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
Financované Európskou úniou NextGenerationEU

6.2. Dostupné možnosti

V prípade softvéru pre OpenStack, sú tri významné možnosti v tejto oblasti: Ceph, GlusterFS a MinIO, pričom každá z nich má svoje výhody i nevýhody.

Ceph, je široko rozšírená voľba, ktorá vyniká svojou bezproblémovou integráciou s OpenStack-om a ponúka robustné blokové, objektové a súborové úložisko v jednotnom systéme, vďaka čomu je vysoko škálovateľné a odolné. Podobne aj samoregeneračné a samosprávne schopnosti softvéru Ceph výrazne znižujú administratívnu záťaž [35][36].

GlusterFS je obľúbenou možnosťou, vďaka svojej jednoduchosti a použiteľnosti, pričom poskytuje škálovateľný sieťový súborový systém vhodný pre úlohy náročné na dáta. Jeho schopnosť integrovať sa s OpenStack-om prostredníctvom modulov Cinder a Manila na ukladanie blokov, resp. súborov z neho robí univerzálnu voľbu pre rôznorodé pracovné zaťaženia [36][37][38].

MinIO sa špecializuje na vysoko výkonné objektové úložisko kompatibilné so systémom S3. Je to výpočtovo nenáročné, ale výkonné riešenie, ktoré je ideálne pre prostredia uprednostňujúce jednoduchosť a efektívnosť pri spracovaní veľkých objemov neštruktúrovaných údajov. Jeho kompatibilita s rozhraním S3 API umožňuje bezproblémovú integráciu do rôznych cloudových aplikácií, vrátane platformy OpenStack [39].

Každá z možností, Ceph, GlusterFS a MinIO predstavuje jedinečný súbor funkcií. Sumárne zobrazenie daných kritérií a ich naplnenie jednotlivých softvérov je zobrazené v tabuľke nižšie.

Tabuľka 6.1 Porovnanie kritérií softvérov Ceph, GlusterFS a MinIO

Kritériá	Ceph	GlusterFS	MinIO
Free/OpenSource	Áno/Áno	Áno/Áno	Bez podpory/Áno
Podporovaný typ úložiska	Objektové, Blokové, Súborové	Objektové, Blokové, Súborové	Objektové
Škálovateľnosť	Áno	Áno	Áno
Podpora API	RADOS, S3	S3	S3
Snapshot/Klonovanie	Áno	Nie	Neaplikovateľné
OpenStack Driver	Áno	Nie	Nie
Podpora Charms	Plná	Nie	Čiastočná

6.3. Výber softvéru

Po dôkladnom vyhodnotení kritérií a dostupných možností sme sa rozhodli zvoliť si ako softvér pre distribuované úložisko Ceph. Jedným z hlavných dôvodov tohto výberu je, že Ceph je open-source platforma, čo zodpovedá našim preferenciám bezplatných a komunitou podporovaných riešení. Okrem toho je významnou výhodou škálovateľnosť, pretože dokáže efektívne spracovávať dáta v až petabajtovom rozsahu, čím zabezpečuje, že infraštruktúra úložiska môže rásť spolu s budúcimi možnými potrebami bez toho, aby došlo k výraznému zníženiu výkonu. Dôležitým faktorom pri našom rozhodovaní boli aj integrované funkcie redundancie softvéru Ceph. Tieto funkcie sú nevyhnutné na zachovanie integrity a dostupnosti údajov, najmä v distribuovanom prostredí. Schopnosť bezproblémovo replikovať a obnovovať dáta poskytuje odolnosť, ktorá je pre poskytovanie cloudových služieb potrebná. Ďalším dôvodom výberu Ceph-u je jeho natívna integrácia s platformou OpenStack, ktorá je kľúčovou súčasťou katedrovej cloudovej infraštruktúry. Ceph podporuje nástroje OpenStack Charms, ktoré uľahčujú správu a nasadenie v prostredí OpenStack.

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
Financované Európskou úniou NextGenerationEU

V prípade ďalších potenciálnych možností ako napr. GlusterFS, nie je v súčasnosti podporovaný platformou OpenStack, od vydania jej verzie Ocata v roku 2017. Neexistencia podpory pre GlusterFS môže mať za následok problémy s integráciou a spôsobiť potenciálnu nekompatibilitu. Navyše nástroje OpenStack Charms nepodporujú GlusterFS, čo je jedným z najdôležitejších kritérií pre výber softvéru. GlusterFS aj MinIO majú svoje silné stránky, ale pre použitie na katedrovom cloude nie sú najvhodnejšie. Napriek tomu, že sa návrh MinIO zameriava predovšetkým na vysoko výkonné objektové úložisko kompatibilné so systémom S3, neobsahuje vo svojej podstate úroveň redundancie, ktorá je požadovaná. Toto obmedzenie by mohlo predstavovať riziko pre odolnosť a dostupnosť dát. Taktiež kompatibilita s nástrojmi OpenStack Charms je veľmi diskutabilná, nakoľko oficiálne dokumentácie neobsahujú túto variantu.

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
Financované Európskou úniou NextGenerationEU

7. CEPH A JEHO ARCHITEKTÚRA

Ceph je open-source systém distribuovaného úložiska, ktorý spravuje nadácia Ceph Foundation v rámci nadácie Linux Foundation. Príspevky širokého spektra organizácií vrátane spoločností Red Hat a IBM zabezpečujú jeho neustály vývoj a zlepšovanie. Poskytuje škálovateľné a jednotné úložisko pre blokové, objektové a súborové údaje a je navrhnutý tak, aby fungoval na komoditnom hardvéri, čo z neho robí nákladovo efektívne a voči chybám odolné riešenie pre rôzne potreby ukladania dát. Od januára 2023 je firma IBM zodpovedná za vývoj, údržbu a podporu Ceph-u, ktorý bol premenovaný na IBM Storage Ceph pre podnikové – „enterprise“ použitie. Táto verzia obsahuje podnikové funkcie a podporné služby s dohodami SLA a je nasadená pomocou linuxových kontajnerov pre jednoduchšiu správu [41][42].

Každý rok, s cieľovým dátumom v marci, sa uvádza nová stabilná verzia, ktorá je pomenovaná v abecednom poradí. Čísla verzií sa používajú vo formáte x.y.z, kde "x" označuje cyklus vydania (poradie písmena v abecede), "y" označuje typ vydania (0 pre vývoj, 1 pre kandidátske vydania, 2 pre stabilné/opravné vydania) a "z" označuje verzie opráv. Stabilné verzie sa aktualizujú každých 4 až 6 týždňov a sú podporované 2 roky [43].

Podľa spoločnosti HG Insights, používajú Ceph ako úložisko aj významné firmy akými sú NVIDIA, logistický gigant Maersk Line, alebo telekomunikačná firma Ericsson [44].

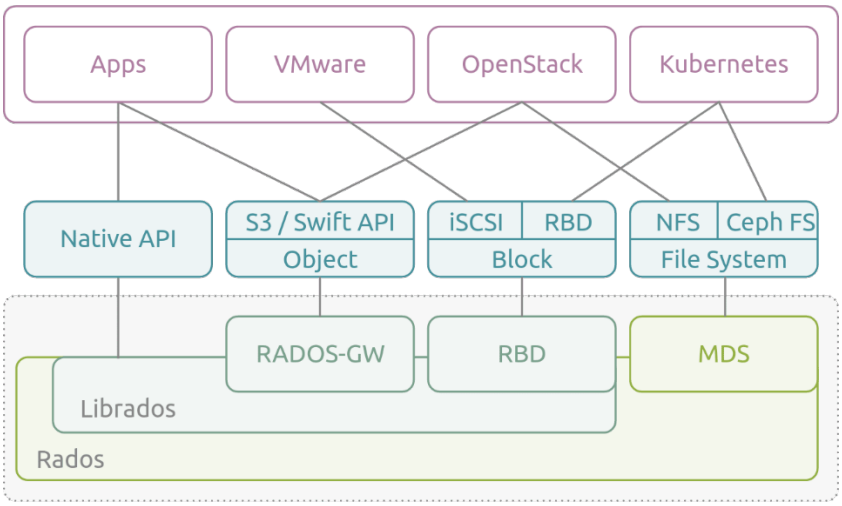
7.1. Kľúčové komponenty softvéru Ceph

Architektúra Ceph sa skladá z niekoľkých kľúčových komponentov, z ktorých každý zohráva významnú úlohu vo funkčnosti a výkonnosti systému. Tieto komponenty spolupracujú, aby poskytli riešenie úložiska, ktoré možno ľahko škálovať a prispôsobovať potrebám používateľa, od terabajtov po exabajty a ďalej bez akéhokoľvek narušenia. V nasledujúcich podkapitolách vysvetlíme ako jednotlivé komponenty pracujú, akú majú úlohu v **klastri** a aké poskytujú výhody [40].

7.1.1. Ceph klaster

Základom celého systému Ceph je klaster. Klaster sa skladá z node-ov, ktoré tvoria jednotlivé zariadenia – servery, na ktorých je nainštalovaný softvér Ceph. Na node-och sa následne nasadzujú jednotlivé komponenty klastra. Vytvorenie klastra Ceph je zložitý proces, ktorý integruje rôzne komponenty a vytvára robustné riešenie úložiska. Sú nimi Ceph Monitor, Ceph Manager, OSD a MDS, pričom každý z nich zohráva v architektúre klastra jedinečnú úlohu. Orchestrácia týchto prvkov umožňuje efektívnu správu blokového úložiska, súborového systému a objektového úložiska v rámci distribuovaného sieťového prostredia. Spolupráca týchto komponentov vytvára komplexný a prispôsobivý systém ukladania dát, ktorý dokáže splniť zložité požiadavky na ukladanie dát v súčasných dátových prostrediach [40].

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
Financované Európskou úniou NextGenerationEU

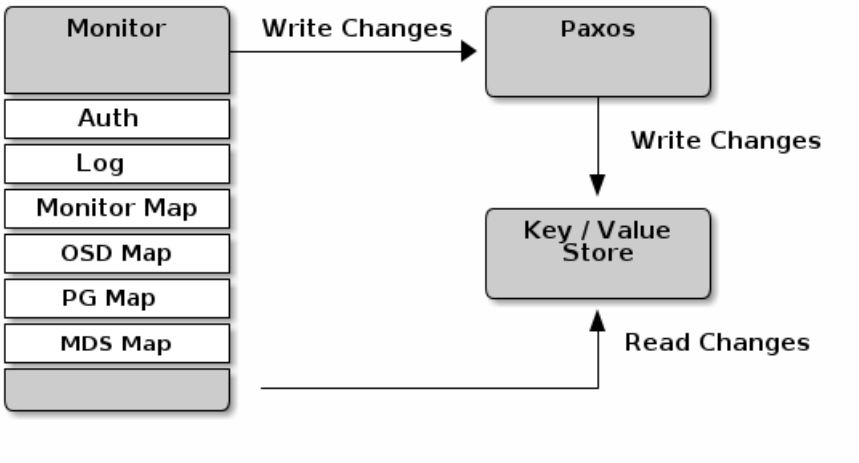


Obrázok 7.1 Architektúra klastra Ceph [40]

Obrázok Obrázok 7.*1* zobrazuje architektúru klastra Ceph, jeho komponenty a flexibilitu využitia. Poskytuje široké možnosti pre komunikáciu s klientami, či už vo forme API rozhraní, alebo pomocou gateway-ov, ktorými poskytuje úložisko pre iné platformy, akými sú VMware, Kubernetes alebo práve OpenStack.

7.1.2. Ceph Monitor

Ceph Monitor (označovaný aj ako „*ceph-mon*“) je dôležitou súčasťou pri udržiavaní funkčného stavu a stability klastra Ceph. Jeho hlavnou funkciou je poskytovať spoľahlivý a aktuálny pohľad na stav klastra a udržiavať mapy klastra. Konkrétne sú to mapy monitora, administratívne mapy, mapy OSD, mapy MDS a mapy CRUSH, ktoré sú nevyhnutné pre koordináciu a efektívnu správu klastra. Vďaka tomu sa klaster môže prispôsobiť zmenám, napríklad pridaniu alebo odstráneniu node-ov bez prerušenia poskytovania služieb. Monitory taktiež zohrávajú kľúčovú úlohu pri správe klastra, vrátane autentifikácie a autorizácie klientov a démonov, čím sa zvyšuje jeho bezpečnosť [40][46][47].



Obrázok 7.2 Zobrazenie úloh Ceph monitora a rozhodovacieho procesu [21]

Obrázok 7.*2* zobrazuje jednotlivé úlohy monitora, ktorými zabezpečuje hladký chod klastra s pomocou konsenzuálneho algoritmu Paxos. Tento algoritmus umožňuje monitorom dohodnúť sa na stave klastra a prijímať rozhodnutia spoločne, čím sa zvyšuje odolnosť voči poruchám, a tým aj spoľahlivosť klastra. Práve konsenzuálnosť algoritmu zaručuje, že klaster

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
Financované Európskou úniou NextGenerationEU

dokáže odolávať zlyhaniam a zachováva integritu aj dostupnosť dát tým, že pre prijatie rozhodnutia je potrebné, aby viac ako polovica monitorov bola v zhode [40][47].

Využitie viacerých monitorov zaručuje, že klaster zostane funkčný aj v prípade zlyhania takmer polovice celkového počtu monitorov. Monitory uľahčujú škálovateľnosť klastra, čo umožňuje prijať viac dát a klientov bez výrazného zvýšenia nákladov na správu. Mechanizmus konsenzu umožňuje klastru udržiavať konzistentný stav aj v prípade rozdelenia siete, alebo zlyhania hardvéru. Dohliadajú na autentifikačný systém cephx, ktorý overuje používateľov a démonov, a chráni pred útokmi napr. typu man-in-the-middle [40][46].

7.1.3. Ceph Manager

Ceph Manager, alebo „*ceph-mgr*“, je jedným z kľúčových komponentov, ktorý poskytuje lepšie možnosti monitorovania a správy klastra. Od verzie 12.x sa stal povinnou súčasťou pre bežnú prevádzku klastra. Démon pracuje popri démonoch monitora, čím poskytuje ďalšie funkcie monitorovania spolu s rozhraniami pre externé systémy monitorovania a správy. Nevyžaduje žiadnu ďalšiu konfiguráciu okrem zabezpečenia svojej prevádzky. Na nastavenie démonov ceph-mgr na monitorovacích node-och sa zvyčajne používajú nástroje na nasadenie, ako napríklad „*ceph-ansible*“, alebo „*cephadm*“, hoci sa nemusia nevyhnutne nachádzať na rovnakých node-och ako monitory [48].

Architektúra manager-a umožňuje rozšírenie jeho funkcií prostredníctvom modulov, ako sú dashboard, prometheus a balancer. Modul dashboard poskytuje zabudované webové rozhranie pre správu, prometheus ponúka možnosti monitorovania a balancer optimalizuje rozdelenie Placement Groups (PGs) medzi OSD démony. Okrem modulov pre prometheus, manager poskytuje aj moduly pre Zabbix, posielanie upozornení, moduly pre orchestráciu a RESTful API rozhranie [48].

Medzi najväčšie výhody manager-a pre klaster Ceph patrí práve modularita a zjednodušené rozhranie pre správu klastra. Tento modulárny prístup umožňuje administrátorom prispôsobiť funkcie pre správu svojim špecifickým potrebám, čím sa optimalizuje výkon a spoľahlivosť klastra ako celku [48].

7.1.4. OSD démony

Ceph Object Storage Daemons (OSDs) – OSD démony sú najdôležitejšími komponentmi architektúry a prevádzky klastra Ceph. Slúžia ako základ pre ukladanie dát, replikáciu, obnovu a monitorovanie stavu klastra. OSD démony alebo aj „*ceph-osd*“, ukladajú dáta v rámci logických pool-ov, zabezpečujú dostupnosť dát a odolnosť systému prostredníctvom svojich úloh pri vyvažovaní dát a kontrole stavu.

Sú nevyhnutné pre dosiahnutie vysokej dostupnosti a odolnosti voči chybám, čo uľahčuje schopnosť efektívne spracovať obrovské objemy dát [35][40].

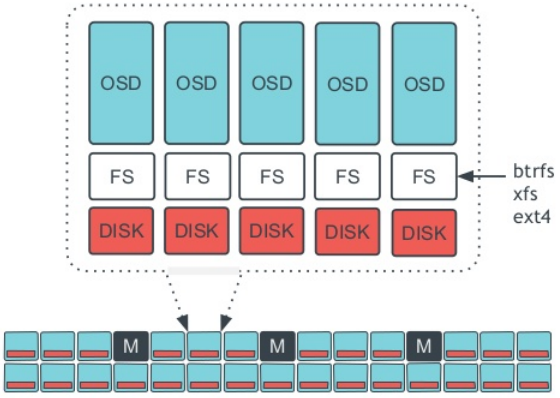
Každý fyzický disk alebo logická partícia určená pre ukladanie dát má zvyčajne vlastný OSD démon. Ukladanie a replikácia dát v klastri sa realizuje práve pomocou OSD démonov, ktoré ukladajú dáta ako objekty v rámci PGs. Na zabezpečenie odolnosti a dostupnosti dát používa Ceph stratégiu replikácie, ktorá zahŕňa replikáciu objektov vo viacerých OSD démonoch. Ich počet je špecifikovaný faktorom replikácie, ktorého hodnota je uvedená v konfigurácii. Distribúciu a správu údajov OSD node-ov efektívne rieši algoritmus CRUSH. CRUSH umožňuje dynamické umiestňovanie objektov a dokáže sa prispôsobiť zmenám v topológii klastra bez toho, aby sa vyžadoval manuálny zásah [40][49].

OSD démony obsahujú aj funkcie monitorovania. V pravidelných intervaloch si posielajú tzv. „*heartbeat*“ správy navzájom aj s monitorom, aby oznámili svoj súčasný stav. Tento monitorovací mechanizmus pomáha klastru rýchlo zistiť a reagovať na prípadné zlyhania OSD démonov. V prípade zlyhania OSD démona Ceph automaticky spustí procesy obnovy dát, ktoré majú za úlohu replikovať dotknuté dáta z iných OSD démonov. V prípade pridania

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
Financované Európskou úniou NextGenerationEU

nových OSD démonov do klastra, Ceph vyvažuje rozloženie dát s cieľom optimalizovať využitie úložiska a jeho výkon [40][49].

Implementácia OSD démonov v klastri poskytuje množstvo výhod v oblasti škálovateľnosti. Pomocou OSD démonov architektúra Ceph umožňuje plynulé navyšovanie kapacity a výkonu úložiska. Pridanie ďalších diskov do klastra zvyšuje celkovú kapacitu úložiska a rozdeľuje pracovné zaťaženie na väčší súbor zdrojov, čím sa zabezpečuje vysoká dostupnosť. Navyše replikácia dát medzi jednotlivými jednotkami OSD démonov zabezpečuje, že žiadny jednotlivý bod zlyhania nemôže ohroziť dostupnosť dát. Z tohoto dôvodu má klaster schopnosť pokračovať vo fungovaní aj v prípade zlyhania niektorých OSD démonov. Ceph využíva replikáciu alebo „erasure coding“ pre zvýšenie odolnosti dát a ochranu pred stratou dát v dôsledku zlyhania hardvéru. Disponujú samoregeneračnými schopnosťami, ktoré umožňujú automatickú obnovu po zlyhaní, redistribúciou dát s cieľom zachovať redundanciu a výkon bez nutnosti zásahu správcu [40][49].



Obrázok 7.3 Zobrazenie OSD démonov na node serveri [24]

7.1.5. MDS a Ceph File System

Metadata Server (MDS) je kľúčovou súčasťou distribuovaného súborového systému CephFS (Ceph File System). Jeho hlavnou úlohou je spravovať metadáta pre operácie so súbormi, ako je otváranie, čítanie, zápis a výpis súborov. Na rozdiel od tradičných systémov ukladania dát, ktoré ukladajú metadáta súborov v rovnakých node-och ako dáta, Ceph používa na správu metadát vyhradené node-y MDS. Táto voľba zlepšuje výkonnosť operácií s metadátami tým, že umožňuje node-om úložiska (OSD) sústrediť sa predovšetkým na ukladanie dát bez dodatočnej réžie správy metadát [51].

Je dôležité poznamenať, že server MDS je relevantný len pri používaní zdieľaného súborového systému, ako napr. CephFS alebo NFS (Network File System). CephFS je distribuovaný súborový systém, ktorý poskytuje vrstvu file storage nad klastrom Ceph. MDS je špeciálne navrhnutý na správu metadát pre systém CephFS, čo z neho robí kritický prvok pre nasadenia, ktoré vyžadujú distribuovaný súborový systém. Pre iné typy úložísk Ceph, ako je blokové úložisko (RBD) alebo objektové úložisko (RGW), MDS servery nie sú potrebné, preto sa bez systému CephFS nevyužívajú [40][51].

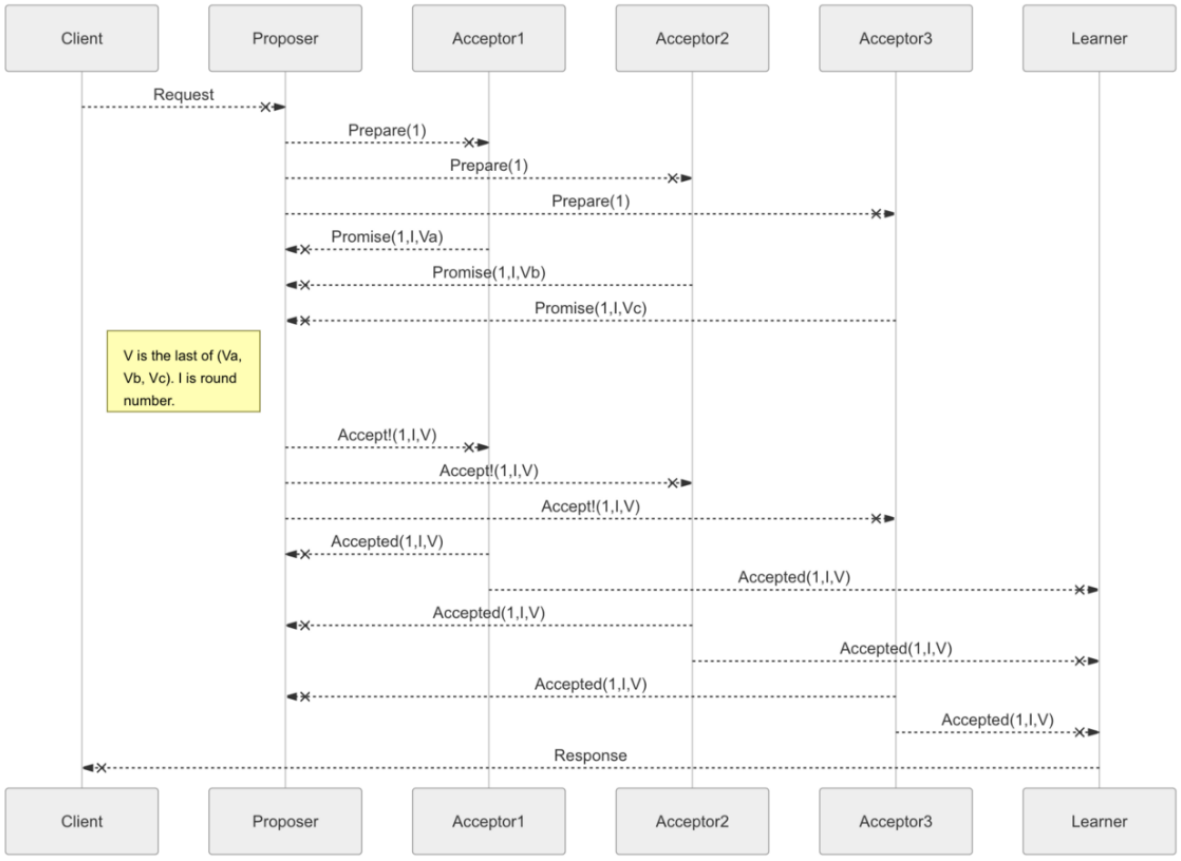
Servery MDS spravujú „*namespace*“ CephFS a koordinujú prístup k samotnému úložisku poskytovaného OSD démonmi. Spracúvajú požiadavky klientov týkajúce sa metadát súborov, čím umožňujú efektívne operácie súborového systému v distribuovanom prostredí. Metadáta sú uložené vo vyhradenej časti v rámci klastra Ceph, čo umožňuje rýchly prístup a správu. Toto nastavenie umožňuje systému Ceph poskytovať klientom rozhranie súborového systému, ktoré je v súlade so štandardmi POSIX a podporovať širokú škálu operácií so súbormi v distribuovanom systéme [52].

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
Financované Európskou úniou NextGenerationEU

Oddelenie ukladania metadát a dát umožňuje klastru s využitím CephFS výrazne optimalizovať operácie založené na súboroch. Oddeleným spracovaním metadát sa znižuje zaťaženie OSD diskov, čo vedie k lepšiemu výkonu a škálovateľnosti operácií s údajmi aj metadátami. Architektúra Ceph-u umožňuje rozširovanie správy metadát pridaním ďalších node-ov MDS, čo je nevyhnutné pri rozsiahlych nasadeniach. Táto schopnosť zabezpečuje, že systém dokáže efektívne spracovať rastúci počet súborov a adresárov. CephFS taktiež podporuje vysokú dostupnosť prostredníctvom nasadenia viacerých serverov MDS v konfigurácii active/standby, čím zabezpečuje nepretržitý prístup k súborom a minimalizuje výpadky v prípade zlyhania MDS. Servery MDS zvyšujú efektívnosť operácií s metadátami prostredníctvom techník, ako je caching a journaling. Journaling je mimoriadne dôležitý na zabezpečenie konzistentnosti a rýchlej obnovy, pretože zaznamenáva zmeny metadát pred ich aplikovaním [40][51][52].

7.2. RADOS

RADOS (Reliable Autonomic Distributed Object Store) je technológia, ktorá tvorí základ distribuovaného systému Ceph a ponúka škálovateľné, spoľahlivé a efektívne riešenia ukladania dát. Je to distribuovaný systém ukladania objektov, ktorý je špeciálne navrhnutý na ukladanie veľkého množstva dát v klastri štandardného hardvéru. RADOS tvorí jadro softvéru Ceph a podporuje rozhrania blokového, súborového a objektového úložiska. Systém je navrhnutý pre vysoký výkon, rozširovateľnosť a spoľahlivosť. Riadi replikáciu, obnovu a distribúciu údajov bez jediného bodu zlyhania [40][52].

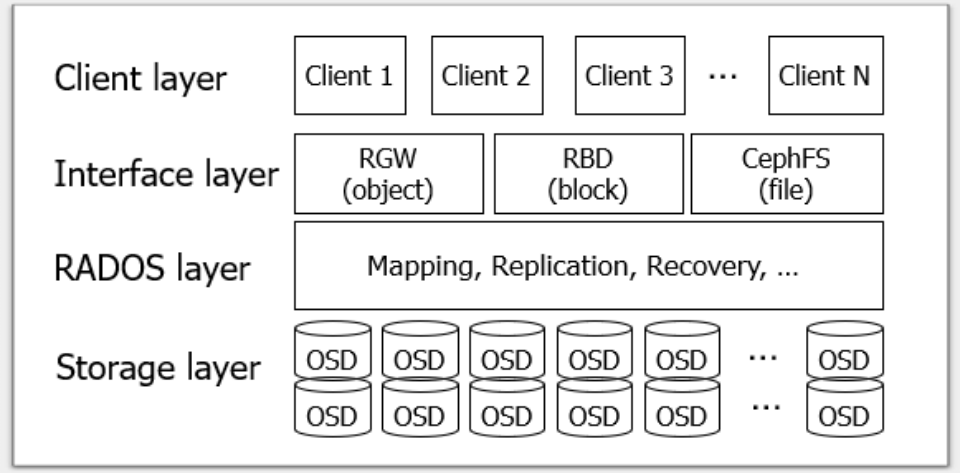


Obrázok 7.4 Princíp dosiahnutia konsenzu pomocou algoritmu Paxos [28]

Nakoľko architektúra klastra Ceph je primárne postavená na RADOS-e s pridaním ďalších funkcií, klastre RADOS podobne pozostávajú z dvoch typov démonov, a to OSD a Monitor. Práve RADOS je technológia, ktorá zabezpečuje takmer všetky známe výhody Ceph-u pre distribuované úložisko. Používa algoritmy CRUSH a Paxos na efektívnu distribúciu

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
Financované Európskou úniou NextGenerationEU

a správu dát v rámci klastra, ktorého princíp je zobrazený na obrázku Obrázok 7.*4* vyššie. RADOS taktiež zabezpečuje trvanlivosť a dostupnosť údajov prostredníctvom replikácie a v neposlednom rade vlastnosť samoregenerácie. V prípade zlyhania node-u, RADOS automaticky prerozdelí dáta do iných node-ov, čím zabezpečí zachovanie replikačného faktora. Tento proces je pre používateľa transparentný a zabezpečuje tak nepretržitú dostupnosť dát [40][53].



Obrázok 7.5 Zobrazenie jednotlivých vrstiev Ceph architektúry [30]

Logické delenie Ceph klastra na vrstvy je zobrazené na obrázku Obrázok 7.*5* vyššie. V klientskej vrstve (Client layer) sa nachádzajú aplikácie, zariadenia alebo platformy, ktoré využívajú distribuované úložisko klastra Ceph. Prístup k dátam zabezpečuje vrstva rozhraní (Interface layer). Ceph ponúka Ceph Rados Gateway (RGW) - rozhranie objektového úložiska, ktoré poskytuje aplikáciám rozhranie RESTful API, Ceph Rados Block Device (RBD) pre ukladanie blokov a image-ov, napr. pre virtuálne zariadenia a virtualizačné platformy a CephFS, ktorý je možné mount-núť ako bežný disk do operačného systému v podobe zdieľaného úložiska. RADOS layer - vrstva RADOS je riadiaca vrstva, ktorá má na starosti všetky vyššie spomínané vlastnosti v predošlom odseku. Poslednou vrstvou je úložná vrstva (Storage layer), ktorá sa skladá zo všetkých OSD démonov v klastri [40].

Ceph ponúka navyše vo vrstve rozhraní aj priamy prístup pre natívne API pomocou knižnice „librados“, ako je zobrazené na obrázku Obrázok 7.*1*. Knižnica librados poskytuje API rozhranie, ktoré umožňuje priamu interakciu s backendom úložiska RADOS. Vytvorené aplikácie sú tak schopné vykonávať operácie priamo na klastri Ceph, čo vedie k efektívnemu ukladaniu a vyhľadávaniu údajov bez potreby použitia klientskej medzivrstvy. Po pripojení môžu aplikácie vykonávať rôzne operácie, ako je čítanie a zápis dát, správa rozšírených atribútov (XATTR) a výpis objektov v rámci úložiska. Librados ponúka podporu asynchrónnych I/O operácií a poskytuje knižnice pre viaceré programovacie jazyky, ako C, C++, Python, Java a PHP [40][57].

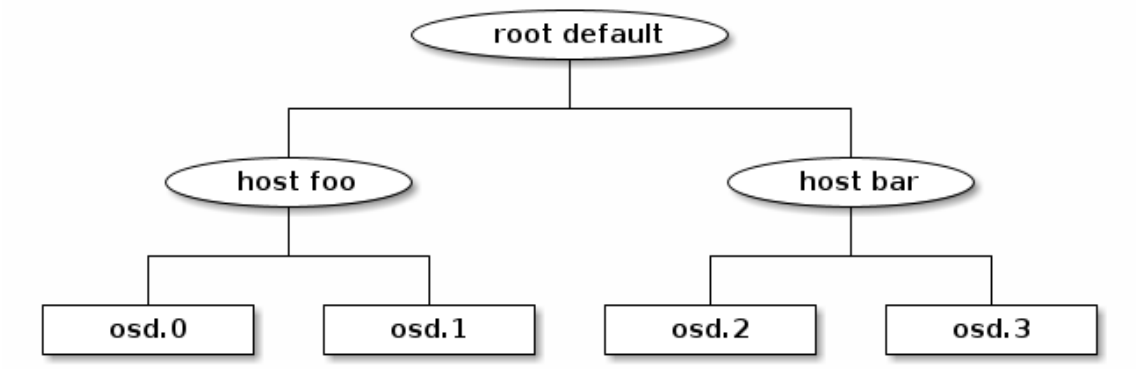
7.3. Algoritmus CRUSH

CRUSH (Controlled Replication Under Scalable Hashing) je pokročilý algoritmus na umiestňovanie dát, ktorý efektívne určuje, ako by sa mali údaje ukladať v úložných node-och klastra. Vďaka tomu sú klastre Ceph vysoko odolné a škálovateľné, pretože distribuujú dáta spôsobom, ktorý maximalizuje dostupnosť a minimalizuje potrebu opätovného vyváženia dát, keď sa klaster zväčšuje alebo zmenšuje. Na rozdiel od tradičných metód prideľovania údajov, ktoré vyžadujú centrálny adresár pre zaznamenávanie pozície jednotlivých dát, CRUSH umožňuje klientom Ceph vypočítať umiestnenie údajov priamo pomocou mapy klastra. Tým sa výrazne znižujú režijné náklady a riziká vzniku potencionálnych bottleneck-ov, čo umožňuje

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
Financované Európskou úniou NextGenerationEU

systému Ceph dosiahnuť spoľahlivosť a výkon pri spracovaní obrovského množstva dát v mnohých node-och úložiska [40][46].

Algoritmus CRUSH využíva hierarchickú štruktúru pozostávajúcu z tzv. „*bucket*-ov“ a OSD pre mapovanie dátových objektov na fyzické úložné miesta. Každá úroveň hierarchie môže obsahovať viacero zariadení alebo iných bucket-ov. Tie následne tvoria stromovú štruktúru odrážajúcu fyzické usporiadanie prostredia úložiska. Táto štruktúra je zakódovaná v tzv. CRUSH mape, ktorá je dodávaná všetkým klientom Ceph a OSD démonom. Príklad CRUSH mapy je možné vidieť na obrázku Obrázok 7.6 [40][58].



Obrázok 7.6 Príklad štruktúry CRUSH mapy [58]

Keď klient Ceph-u potrebuje zapisovať údaje, použije algoritmus CRUSH a najnovšiu mapu CRUSH na výpočet, ktorý OSD by mal uložiť konkrétny dátový objekt. Pri tomto výpočte sa zohľadňuje hierarchia, váhy zariadení a pravidlá replikácie alebo erasure coding pravidlá definované pre dáta, čím sa zabezpečí rovnomerné rozloženie dát v rámci klastra a dodržanie stanovených požiadaviek na redundanciu [40][58].

Distribúciou údajov predvídateľným, ale náhodným spôsobom medzi viacerými OSD, zvyšuje CRUSH dostupnosť údajov a odolnosť voči chybám. Umožňuje klastru dynamicky vyvažovať údaje pri pridávaní alebo odstraňovaní OSD, čo je kľúčové pre rozsiahle systémy ukladania dát, ktoré sa musia škálovať, aby sa prispôsobili rastúcim objemom dát bez narušenia prebiehajúcich operácií [40][46][58].

CRUSH umožňuje klastrom Ceph horizontálne sa rozširovať s minimálnym dopadom na výkon rovnomerným rozložením pracovného zaťaženia na všetky dostupné zdroje. Znižuje latenciu a zvyšuje priepustnosť tým, že poskytuje priamu komunikáciu klienta s OSD bez centralizovanej vyhľadávacej tabuľky. Administrátori majú možnosť definovať vlastné hierarchické štruktúry, ktoré zodpovedajú ich skutočnému rozloženiu hardvéru, čím sa optimalizuje výkon a trvanlivosť dát [40][46][58].

7.4. Pool-y a PGs

Pool-y sú základnými jednotkami, v ktorých sa nachádzajú objekty v rámci klastra Ceph. Zohrávajú kľúčovú úlohu pri orchestrácii distribúcie a odolnosti dát. PGs (Placement Groups) sú kľúčovým prvkom architektúry Ceph, ktorý zefektívňuje organizáciu, prideľovanie a prístup k údajom v distribuovanom prostredí. Tento mechanizmus umožňuje Ceph-u ľahko škálovať pri zachovaní dostupnosti a vyváženej distribúcie dát v rámci klastra. Porozumenie fungovania PGs je nevyhnutné pre pochopenie, ako Ceph dosahuje redundanciu a efektívnosť pri správe dát [40][59][60].

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
Financované Európskou úniou NextGenerationEU

7.4.1. Pool

V klastri Ceph-u sú pool-y logické partície, ktoré obsahujú objekty. Fungujú ako rozhranie pre klientov na ukladanie a načítanie dát. Ceph podporuje dva typy pool-ov: „replicated“ a „erasure coded“. Replicated pool-y poskytujú redundanciu tým, že ukladajú viacero kópií objektu na rôznych OSD. Erasure coded pool-y používajú na ochranu dát paritu, čím poskytujú mechanizmus redundancie, ktorý je efektívnejší z hľadiska úložiska ako replikované pool-y [40][59].

Fungovanie pool-ov je úzko spojené so schopnosťou klastra spravovať a distribuovať dáta. Tie sa distribuujú na OSD využitím PGs a algoritmu CRUSH [14].

Pool-y Ceph-u sú konfigurovateľné, čo umožňuje administrátorom upravovať nastavenia, ako je počet replík objektov, pravidlá CRUSH a parametre súvisiace s výkonom. Umožňujú logické zoskupovanie dát, čo uľahčuje ich správu a prístup k nim. Pre rôzne typy údajov, ako sú obrázky, videá alebo zálohy, možno vytvoriť rôzne pool-y, pričom každý z nich má vlastné nastavenia [40][59].

Je dôležité však pochopiť nasledujúci koncept. Po vytvorení pool-u, Ceph automaticky spravuje distribúciu dát medzi dostupné OSD na základe konfigurácie a algoritmu CRUSH. Preto sú pool-y nakonfigurované tak, aby riadili stratégie replikácie a umiestňovania dát, a nie aby zahŕňali konkrétne node-y. V predvolenom nastavení sa všetky OSD podieľajú na ukladaní dát pre všetky pool-y. OSD démony nie sú konkrétne priradené ku konkrétnemu pool-u. Namiesto toho algoritmus CRUSH určuje, ako sa údaje distribuujú medzi OSD na základe konfigurácie pool-u a topológie klastra [59].

V klastri Ceph, rozhodnutie o tom, kde bude objekt uložený – konkrétne, v ktorom pool-e, sa vykonáva v čase zápisu objektu do klastra. Toto rozhodnutie je založené na aplikácii alebo používateľovi, ktorý špecifikuje cieľový pool pre dáta [40].

7.4.2. PG

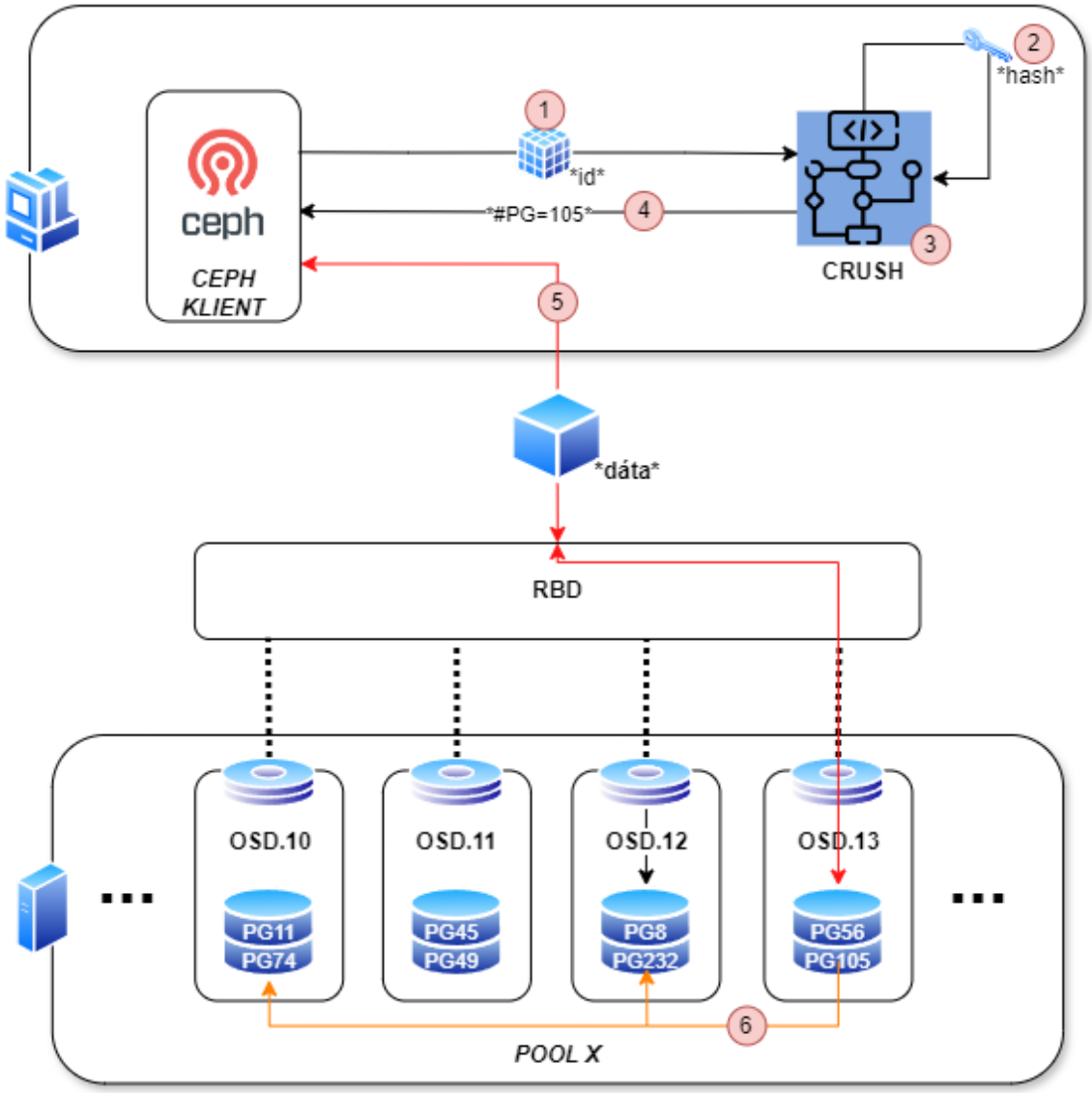
Placement Groups (PGs) sú logické partície v rámci Ceph pool-ov. Združujú objekty, čím zefektívňujú prístup k objektom. PGs znižujú réžiu spojenú so sledovaním umiestnenia objektov a metadát, čo je čoraz náročnejšie, keď počet objektov narastá na milióny. Každá PG je potom priradená k jednotlivým démonom OSD v rámci pool-u, pričom jedna PG slúži ako primárna (na koordináciu operácií zápisu alebo čítania pre klienta) a ostatné sekundárne pre redundanciu dát [40][60].

Proces sa začína tým, že klient Ceph identifikuje PG, do ktorej objekt patrí, zvyčajne na základe jeho mena alebo iného jedinečného identifikátora. Táto identifikácia sa vykonáva pomocou funkcie hash-ovania. Výsledkom hash-ovacej funkcie je číslo a toto číslo sa použije na výber jednej z dostupných PGs. Tento proces je navrhnutý pre nahradenie klasického spôsobu ukladania metadát napr. stromovou štruktúrou. Ak by sme zobrali do úvahy situáciu, kedy chceme nájsť jeden konkrétny objekt bez predošlého záznamu v cache medzi miliónom objektov, určenie presného miesta, kde sa objekt nachádza, je mimoriadne časovo náročné, a tým sa znižuje výkon klastra. Použitím PGs sa minimalizuje čas potrebný na lokalizovanie umiestnenia objektu, keďže na základe vypočítanej hash-e dokáže algoritmus CRUSH takmer okamžite určiť OSD a PG, v ktorom je objekt uložený. Na základe týchto údajov algoritmus CRUSH nasmeruje klienta na konkrétne OSD, kam dáta zapíše alebo dáta prečíta [61].

Ceph minimalizuje výpočtovú a pamäťovú réžiu zoskupovaním objektov do PGs. Tie umožňujú efektívnu distribúciu a replikáciu dát v potenciálne veľkom počte OSD démonov. Odolnosť a dostupnosť dát je zabezpečená replikáciou dát na viacerých OSD v rámci PGs, a to aj v prípade zlyhania OSD. Ceph má funkciu nazývanú automatické škálovanie PGs, ktorá dokáže upraviť počet PGs v poole na základe využitia úložiska a definovaných pravidiel. To pomáha udržiavať výkon a využitie zdrojov bez manuálneho zásahu. Automatické škálovanie

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
Financované Európskou úniou NextGenerationEU

berie do úvahy celkovú kapacitu úložiska, rozloženie dát medzi rôzne úložné zariadenia a očakávanú veľkosť dát, aby podľa potreby upravilo počet PGs [40][62].



Obrázok 7.7 Princíp zápisu/čítania dát s použitím klienta

Pre priblíženie procesu, ako funguje zápis, alebo čítanie dát použijeme Obrázok 7.7. Proces sa skladá zo šiestich krokov. V prvom kroku klient pošle identifikátor algoritmu CRUSH. Ten z identifikátora v druhom kroku vypočíta hash. V treťom kroku sa z vypočítanej hash-e vypočíta podľa predurčeného algoritmu číslo PG. Štvrtý krok je odoslanie čísla PG klientovi. Keďže klient disponuje celou aktuálnou mapou CRUSH, v piatom kroku na základe čísla PG nadviaže spojenie priamo s príslušným OSD z vybraného pool-u, na ktorom sa PG nachádza. Podľa operácie sa buď dáta zapíšu alebo prečítajú. Posledným šiestym krokom je v prípade zápisu dát replikácia na sekundárne OSD, resp. do ich príslušných PGs.

Ďalšie informácie o nasadení a testovaní riešenia Ceph pre Openstack sú dostupné v podpornom dokumente s názvom „Distribuované úložisko“.

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
Financované Európskou úniou NextGenerationEU

8. IMPLEMENTÁCIA ZDIEĽANÉHO SÚBOROVÉHO SYSTÉMU V PROSTREDÍ OPENSTACK

V rámci tejto časti upriamime pozornosť na to, akým spôsobom je možné implementovať zdieľaný sieťový súborový systém do prostredia cloudu na platforme OpenStack, a to s využitím modulu Manila.

8.1. Úložiská v prostredí platformy OpenStack

Po výbere CephFS ako najvhodnejšieho riešenia pre infraštruktúru OpenStack na KIS, ako bolo spomenuté už v predchádzajúcich kapitolách, je nevyhnutné zabezpečiť, aby sa takto nasadené riešenie dalo efektívne a jednoducho spravovať. V rámci platformy OpenStack sa využívajú dva hlavné projekty pre úložiská, ktorými sú Cinder a Swift.

Cinder poskytuje API slúžiace na riadenie blokového úložiska, ktorý poskytuje vysokovýkonné pripojenie pre virtuálne stroje a iné dáta, avšak volume (pamäťový disk) môže byť prístupný len jednému takémuto virtuálne stroju v danom čase. O takéto obmedzenie ide z toho hľadiska, že v sebe nezahŕňa žiadne mechanizmy zámkov či synchronizácie prístupu [10].

Modul Swift zase poskytuje objektové úložisko, ktoré je obzvlášť vhodné na ukladanie veľkých binárnych objektov ako sú obrázky, videá, či rôzne zálohy. Nevýhodou však je, že takto uložené objekty v ňom sú nemenné, teda po jeho uložení ho už nie je možné upraviť či modifikovať po častiach. Takto zmeniť objekt je možné len jeho úplným prepísaním, a preto nie je vhodný pre správu takých dát, ktoré sa často menia alebo sú predmetom transakčných operácií, špecificky databázy. Navyše, neodporúča sa ani na ukladanie malých objektov, pretože v sebe obsahuje implementované metódy ich ochrany, čo by v prípade častého narábania s nimi, sa odzrkadlilo na rýchlosti operácií [10].

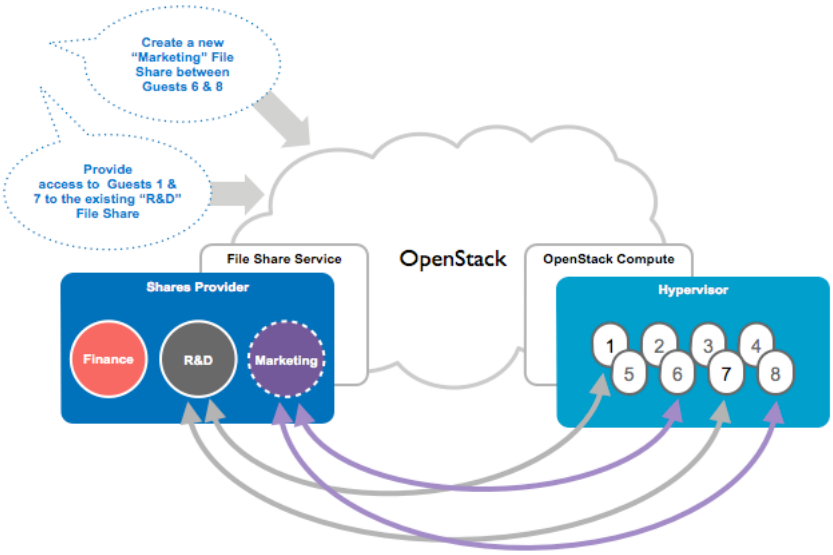
Z týchto dôvodov vznikol modul Manila, odvodený od modulu Cinder a poskytuje veľmi podobné riadiace rozhranie pre zdieľané súborové systémy [10].

8.2. Manila ako služba pre správu zdieľaných sieťových súborových systémov v prostredí OpenStack

Manila je služba špecificky navrhnutá na správu zdieľaných sieťových súborových systémov v prostredí OpenStack. Jej hlavnou funkciou je poskytovať prístup modulu Nova (zodpovednému za vytváranie a riadenie virtuálnych inštancií) k zdieľanému súborovému úložisku. Tým umožňuje virtuálnym inštanciám čítať a zapisovať dáta na zdieľané úložisko, čo prináša výhody centralizovaného ukladania súborov [10][11].

Služba Manila poskytuje komponenty orchestrácie zodpovedné za proces vytvárania a priraďovania zdieľaných súborových systémov. Veľkou výhodou tohto modulu je to, že poskytuje len vrstvu riadenia, ktorá má za úlohu administratívu, zatiaľ čo samotná interakcia medzi úložiskom a inštanciami je riadená na úrovni backendu. Táto funkcionalita je implementovaná ako súbor API, príkazového riadku (CLI), a tiež integrácie do Horizon (OpenStack dashboard – grafické používateľské rozhranie) [10][12].

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
Financované Európskou úniou NextGenerationEU



Obrázok 8.1 Priebeh komunikácie medzi modulom Manila a Nova inštanciami [12].

V rámci tejto služby sa využívajú rôzne koncepty a termíny, najdôležitejšími sú:

- **Share** – Jedná sa o konkrétnu inštanciu súborového systému – zdieľanie, ktorá je priradená k virtuálnemu počítaču alebo serveru a môže obsahovať dáta prístupné viacerým inštanciám súčasne s istotou, že prístup jednotlivých inštancií bude riadený a všetky inštancie pristupujú k rovnakým súborom [10][12][13].
- **Share Access Rule** – Predstavujú bezpečnostné pravidlá/pravidlá prístupu, ktoré definujú kto a akým spôsobom môže pristupovať k danému zdieľaniu. Fungujú podobne ako ACL [10][13].
- **Security Service** – Slúži na overovanie identity používateľov, pomocou externých služieb ako sú „LDAP“ (Lightweight Directory Access Protocol) alebo „Micrsosoft Active Directory“. Takto budú môcť administrátori jednoducho definovať skupiny používateľov, ktoré budú mať prístup k danému zdieľaniu [10][12].
- **Share Network** – Reprezentuje sieťovú konfiguráciu priradenú k danému zdieľaniu. Okrem špecifikácie samotnej siete, či podsiete slúži aj na implementáciu „multi-tenancy“ (spôsob oddelenia tej istej infraštruktúry pre viacerých používateľov tak, aby mali izolované a bezpečné prostredie bez vzájomného ovplyvňovania sa) pomocou VLAN a VXLAN, vďaka čomu sú dáta naozaj oddelené a neprístupné pre používateľov z inej siete [10][12].

Medzi ďalšie relevantné termíny pri implementácii Manily patria ešte:

- **Snapshot** – Predstavuje statickú kópiu existujúceho zdieľania v danom čase. Slúži predovšetkým na účely zálohovania a prípadného obnovenia dát [12][13].
- **Storage Pools** – Sú logické skupiny úložných prostriedkov, ktoré efektívne umožňujú Manila alokovať a spravovať úložisko a súčasne poskytujú jej používateľom flexibilitu pri výbere miesta na uloženie svojich dát [12].
- **Share Type** – Umožňuje administrátorom si osobitne nadefinovať kritériá pre rôzne úrovne služby. Napríklad vytvorenie dvoch tried zdieľaní, kde jedno predstavuje základné jednoduché zdieľanie pre bežnú obsluhu klientov, zatiaľ čo druhé bude definované, ako výkonnejšie, pre osobitné účely [12].

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
Financované Európskou úniou NextGenerationEU

- **Share Server** – Tvorí logickú entitu, ktorá hostí zdieľania priradené k určitej zdieľanej sieti. Tento server môže byť buď konfigurácia v rámci úložného systému, alebo iné logické zdroje poskytované platformou OpenStack. Share server je tiež zodpovedný za určovanie správnych IP adries na exportovanie zdieľaní v rámci príslušnej siete [12].

Samotná služba môže bežať v rôznych konfiguráciách, a to buď na jednom alebo aj viacerých uzloch v klastri a je schopná naraz poskytovať podporu pre viacero rôznych backendov. V rámci tejto diplomovej práce sme sa však rozhodli len pre jeden backend, a tým je CephFS, ktorý bude slúžiť ako zdieľané úložisko súborov [12].

8.3. Ciele a výhody použitia služby Manila

Manila predstavujúca službu s otvoreným zdrojovým kódom pre potreby zdieľaných sieťových súborových systémov v prostredí OpenStack si kladie niekoľko základných cieľov, ktorými sú:

- **Modulárna architektúra pre jednoduché rozširovanie** – Služba využíva komponentovo orientovanú architektúru, čo vo svojej podstate umožňuje rýchle pridávanie nových funkcií a ďalších rozšírení. Vďaka tomu je celá služba prispôsobivá podľa požadovaných kritérií daného nasadenia [11]
- **Škálovateľnosť pre veľké pracovné záťaže** – Manila je navrhnutá tak, aby bolo schopná spracovať aj náročné úlohy predstavujúce vysokú záťaž systému. Podporuje horizontálnu aj vertikálnu škálovateľnosť [11].
- **Odolnosť proti výpadkom** – V rámci služby sú jednotlivé procesy od seba oddelené, čím sa zabraňuje šíreniu zlyhaní medzi rôznymi časťami systému, prechádza sa tým tzv. reťazovým výpadkom, ktoré by mohli mať za následok znefunkčnenie celej služby [11].
- **Bezpečné zdieľanie súborov s riadením prístupu** – Nakoľko v rámci Manily sa ponúkajú až dve možnosti riadenia prístupu používateľov, a to pomocou Share Access Rule a Security Service, patrí medzi jej najväčšie výhody práve funkcionalita možnosti kontroly prístupu používateľov k zdieľaným súborom. Všetky súbory sú tak v bezpečí a prístupné len tým serverom a používateľom, ktoré predtým administrátori povolili [11].

8.4. Kľúčové komponenty architektúry služby Manila

Samotná služba sa skladá zo 4 hlavných komponentov, a to Manila-api, Manila-data, Manila-scheduler a Manila-share. Inštalácia prvých troch je identická pre rôzne spôsoby nasadenia, tá posledná, je však závislá od zvoleného typu backendu ako úložiska, ktorým je pre nás CephFS [12].

Manila-api

Komponent Manila-Api predstavuje niečo ako vstupný bod služby zdieľaného sieťového súborového systému v OpenStacku. Poskytuje RESTful API, prostredníctvom ktorého je možné vytvárať, upravovať a mazať vytvorené inštancie zdieľaného súborového systému. Teda úlohou tohto komponentu je prijímať API požiadavky od klientov/používateľov alebo iných služieb na správu zdieľania a odosielať ich na spracovanie backendu, ako je nami zvolený CephFS. Navyše, tento komponent sa stará aj o autentifikáciu a autorizáciu požiadaviek prostredníctvom modulu Keystone [14].

Manila-data

Hlavnou úlohou tohto komponentu je spracovávať dátové operácie, medzi ktoré patrí kopírovanie, migrácia zdieľaných súborov či zálohovanie. Výhodou oddelenia týchto náročných operácií a ich vyčlenenie do samostatne služby má za následok lepšiu škálovateľnosť jednotlivých komponentov modulu Manila [14][15][16].

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
Financované Európskou úniou NextGenerationEU

Manila-scheduler

Nasledujúci komponent zodpovedá za výber vhodného backendu pre nové zdieľanie. Keď používateľ alebo služba odošle požiadavku na vytvorenie nového zdieľania prostredníctvom manila-api, tá ju odovzdá manila-scheduleru. Ten následne analyzuje všetky nakonfigurované backendy, kde pri následnom rozhodovaní o tom, ktorý zvolí, využíva rôzne konfigurovateľné filtre a váhy, ktoré zohľadňujú špecifické atribúty ponúkaných backendov. Týmito váhami sú napríklad dostupná kapacita, zóna dostupnosti či iné. Po zvolení vhodného backendu, komponent manila-scheduler odošle požiadavku na manila-share, ktorý ďalej pokračuje v riadení vytvárania zdieľania [14][15].

Manila-share

Komponent manila-share je zodpovedný za samotnú tvorbu, správu a exportovanie zdieľaní na príslušnom backende. Predstavuje trvalý čitateľný a zapisovateľný súborový systém, ku ktorému môže pristupovať rôzny počet klientov [15][16].

V tejto implementácií ma definovaný protokol, veľkosť a zoznam prístupových práv, ktorý určuje, ktorí klienti k nemu majú prístup. Služba manila-share komunikuje s úložiskom pre potreby zabezpečenia a správy pre konkrétne zdieľanie a manipulácie s ním [16].

8.5. Interakcia komponentov služby Manila

Komunikácia medzi jednotlivými komponentami prebieha spravidla pomocou fronty správ, kde sa väčšinou používa „message broker“ (program slúžiaci na zdieľanie informácií medzi aplikáciami) ako je RabbitMQ. Okrem toho služba pracuje s databázou, v ktorej si ukladajú metadáta o zdieľaných súborových systémoch, pravidlách prístupu a stave jednotlivých operácií [17].

Z vonkajšieho pohľadu klient komunikuje so službou cez python-manilaclient, a to prostredníctvom príkazového riadku alebo ako súčasť skriptov a programov využívajúcich Python knižnicu. Tá potom prijatú požiadavku odosiela cez REST API na API server, ktorý ju presmeruje na Auth manager-a, čo spravidla býva modul Keystone zabezpečujúci autentifikáciu. Po overení identity používateľa komponent API požiadavku spracuje, uloží potrebné informácie do databázy a následne ju odošle cez správnu frontu pomocou AMQP „Advanced Message Queuing Protocol“ protokolu ďalšiemu komponentu, v tomto prípade manila-scheduler [17].

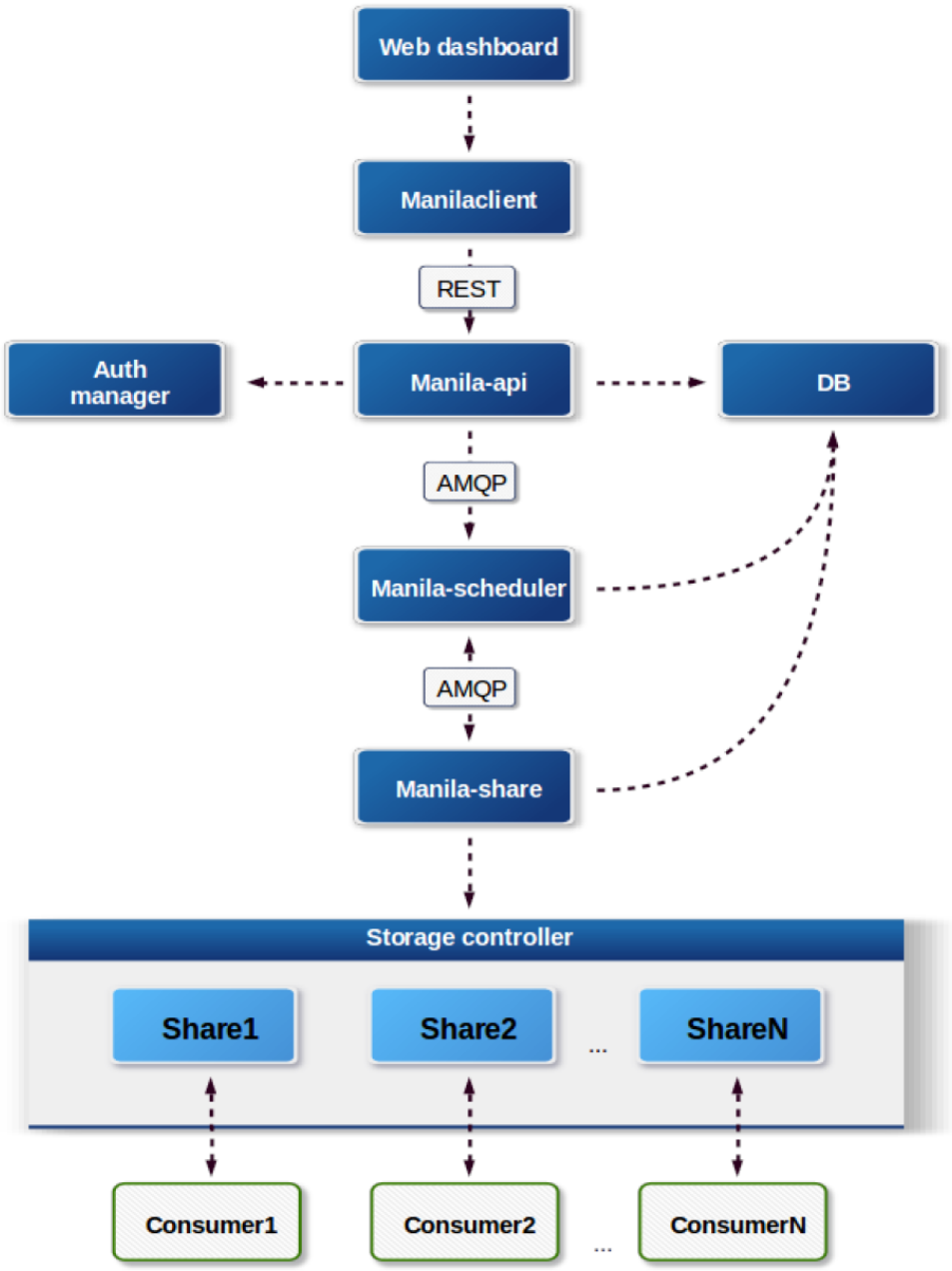
Keďže úlohou manila-scheduler je rozhodnúť, ktorý backend má byť využitý ako úložisko príslušného zdieľania, tak vstúpi do databázy, z nej získa aktuálne informácie o dostupných backendoch, ako je napríklad ich kapacita či stav a na základe tých potom vyberie ten najvhodnejší. Tento výber potom vloží do správy a odošle ho komponentu manila-share na spracovanie [17].

Komponent manila-share túto správu prevezme a vykoná príslušnú úlohu, ktorou môže byť ako už bolo spomenuté vytváranie, mazanie, či aktualizácia príslušných zdieľaní. Manila-share pritom priamo komunikuje so zvoleným backendom. V závislosti od typu vybraného backendu sa používajú príslušné Manila ovládače, ktoré sprostredkúvajú komunikáciu a prevádzku na strane úložiska [17].

V poslednom kroku sú novo vytvorené zdieľania exportované a sprístupnené pre koncových používateľov, ktorí k nim pristupujú pomocou backendom definovaných protokolov [17].

Celý priebeh komunikácie opísanej v tejto kapitole je možné vidieť na nižšie uvedenom obrázku:

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
Financované Európskou úniou NextGenerationEU



Obrázok 8.2 Priebeh komunikácie Manila služby spolu s ostatnými aplikáciami [17].

V rámci prostredia OpenStack ide z pohľadu používateľa o bežné sieťové úložisko, ktoré si môže mountnúť (pripojiť) do svojho operačného systému, v rámci napríklad virtuálnej inštancie, rovnako ako akýkoľvek iný sieťový disk, pričom takéto vytvorené zdieľané systémy môžu byť nakonfigurované s rôznymi úrovňami prístupových práv (read-only, read-write) [17].

Presnejšie a technické informácie o nasadení a skriptoch, ktoré nasadia komponent Manila do systému OpenStack pomocou Juju Charms sa nachádza v podpornom dokumente „podklad pre V3 v rámci KPB1_Nasadenie_Manila_bundle.docx“.

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
Financované Európskou úniou NextGenerationEU

9. MONITORING V PROSTREDÍ CLOUD COMPUTING-U

Na úvod je vhodné definovať, čo to vlastne pojem *Cloud computing* reprezentuje, V tejto kapitole sa pokúsime identifikovať motiváciu za monitorovaním takéhoto systému a rozdelíme rôzne pohľady.

9.1. Motivácia pre monitorovanie

Záujem o monitorovanie CC prostredia má ako poskytovateľ, tak aj používateľ takejto platformy. Je zrejmé, že prevádzkovateľ služby potrebuje detegovať zlyhania, ako na fyzickej, tak vo virtuálnej infraštruktúre. Jeho záujmom by taktiež malo byť optimalizovať chod infraštruktúry [14].

Keďže vzťah medzi poskytovateľom a zákazníkom by mal byť formálne zadefinovaný, v tomto prípade v zmluve o úrovni služby (SLA), monitorovací systém by mal vedieť upozorniť vlastníka systému na prípadné problémy, ktoré môžu ovplyvniť výkon služby, a tým spôsobiť finančné škody.

Zákazník má záujem monitorovať jemu poskytnuté zdroje, a to najmä aby mal prehľad, či potrebuje pridelené množstvo, keďže väčšie množstvo pridelených prostriedkov znamená vyššie náklady. Na druhej strane musí vedieť, či jemu pridelené zdroje stále postačujú požiadavkám jeho projektu, a či nie je potrebné ich navýšiť [14].

Existuje veľké množstvo procesov, ktoré sú pre beh CC platformy kritické, pričom pre každý z nich je ich monitorovanie kľúčovou úlohou. Dané procesy, ako aj rolu ich monitoringu uvádzame nižšie [15].

9.1.1. Kapacita a plánovanie zdrojov

Pred masovým adoptovaním CC bolo pre prevádzkovateľov a vývojárov aplikácii jednou z najväčších výziev plánovanie a manažovanie fyzických zdrojov, na ktorých daná aplikácia fungovala. Aby dokázali zaručiť istú kvalitu služby, potrebovali kvantifikovať, aké množstvo prostriedkov potrebujú pre hladký beh procesov. Vo všeobecnosti platí, že takýto odhad je možné vytvoriť na základe testovania, monitorovania a štatistickej analýzy, no reálne hodnoty môžu byť nepredvídateľné a vysoko variabilné. V tomto je pre poskytovateľov aplikačných služieb CC veľkou výhodou, keďže celý tento proces monitorovania a spravovania infraštruktúry prechádza na prevádzkovateľa CC systému. Tu sa dostávame k faktu, ktorý uvádzame aj vyššie v kapitole 9.1. Monitorovanie priľahlej infraštruktúry je z pohľadu poskytovateľa CC služieb kritické, pokiaľ chce sledovať a predpovedať vývoj parametrov zastúpených v procese zachovania kvality služby (QoS). Na základe výstupov z takéhoto monitorovacieho systému by mal vedieť správne naplánovať rozvoj fyzickej infraštruktúry a poskytovaných zdrojov tak, aby boli rešpektované SLA, uzatvorené zo všetkými zákazníkmi [15].

9.1.2. Kapacita a manažovanie zdrojov

Základným predpokladom funkčného monitorovacieho systému je schopnosť zachytiť aktuálny stav CC platformy. Hlavnou doménou v CC je virtualizácia, ktorá zakrýva heterogenitu priľahlej fyzickej infraštruktúry, a tým prináša vyššiu úroveň zložitosti pre poskytovateľa, ktorý tým pádom potrebuje manažovať ako fyzickú, tak aj virtuálnu infraštruktúru. Virtuálne zdroje môžu migrovať medzi fyzickými zariadeniami, čo môže spôsobovať variabilnosť dostupných zdrojov na jednotlivých fyzických zariadeniach. Správne nasadený monitorovací systém dokáže administrátora upozorniť, ak by dochádzalo k nedostatku výpočtových zdrojov na niektorom z uzlov. Okrem toho je pre veľa zákazníkov dôležitý aspekt dostupnosti služby. Monitorovanie dokáže odhaliť anomálie v chode systému, a preto ho považujeme za potrebný v kontexte zaručenie čo najväčšej dostupnosti [15].

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
Financované Európskou úniou NextGenerationEU

9.1.3. Manažovanie dátového centra

Cloudové služby bývajú poskytované na podklade veľkých dátových centier, ktorých riadenie je dôležitou úlohou pre bezproblémový chod komplexného systému. Aktivity riadenia dátového centra vieme rozdeliť do dvoch základných úloh:

- **Monitorovanie** – Sledovanie požadovaných hardvérových a softvérových metrík.
- **Dátová analýza** – Spracovanie nazbieraných metrík za účelom vyhodnotenia stavu systému.

Je dôležité, aby takýto monitorovací systém umožňoval veľké množstvo operácií v reálnom čase, keďže v dátovom centre je predpoklad výskytu veľkého množstva prvkov infraštruktúry. V tomto kontexte je hlavná motivácia pre monitoring úspora energie, čiže optimalizácia behu fyzických zariadení [15].

9.1.4. Manažment SLA

Monitoring je povinnou a nenahraditeľnou súčasťou procesu overovania dodržiavania SLA, napríklad keď sa vykonávajú audítorské aktivity pre zistenie dodržiavania regulácií (napr. v prípade, ak sa služba ponúka štátnym orgánom). Dáta z monitorovacieho systému môžu poskytovateľovi pomôcť naformulovať SLA viac realisticky a navrhnúť udržateľný model fakturovania [15].

9.1.5. Fakturovanie

Ako už bolo spomenuté vyššie, jednou zo základných charakteristík CC je poskytovanie služieb na báze „plať za to, čo použiješ“. To umožňuje zákazníkovi platiť proporcionálne k využívaniu informačných zdrojov. Základným predpokladom pre fakturáciu takýmto spôsobom je existencia procesu, ktorý sleduje množstvo využitých zdrojov zákazníkov. Tento proces môže byť, buď samostatný, alebo spojený v jedno s monitorovacím riešením [16].

9.1.6. Zisťovanie problémov

Infraštruktúra CC býva často komplexná a predstavuje výzvu v prípade presného určenia pôvodu vzniknutého problému. Pri výskyte problému je potrebné analyzovať niekoľko možných komponentov, ktoré mohli zlyhať. Ďalší fakt, ktorý pridáva na zložitosti, je aj to, že samotné komponenty sa môžu skladať z niekoľkých vrstiev (napr. fyzická alebo virtuálna vrstva, operačný systém (OS) hostiteľa alebo virtuálneho stroja (VM) a pod.). Zrozumiteľný a detailný monitorovací systém dokáže poskytovateľovi pomôcť rýchlejšie, a presnejšie diagnostikovať vzniknutý problém. Okrem toho by si mal zákazník vedieť overiť, či sa prípadná chyba vyskytuje na jeho, alebo prevádzkovateľovej strane [15], [16].

9.1.7. Manažovanie výkonu

V praxi platí, že niektoré výpočtové uzly v CC poskytujú väčší výkon ako iné. Na bežného zákazníka toto nemá zásadný vplyv, keďže záťaž sa zvykne rovnomerne rozdeľovať a pokles výkonu nebýva zväčša významný. Môžu však existovať zákazníci, ktorí poskytujú s využitím CC kritické služby. Pre nich je každá zmena výkonu významná, pričom dostupnosť výpočtových zdrojov je kritická. Pri takýchto službách zvykne daný zákazník využívať niekoľko rôznych poskytovateľov, pričom si vyberá toho aktuálneho (na ktorom bude v tom momente bežať služba) na základe poskytnutého, vnímaného výkonu. Zákazník má v tomto prípade záujem monitorovať jeho vnímaný výkon služby, pričom poskytovateľ mu chce poskytnúť čo najspoľahlivejšie zdroje. Pre obidve strany je v tomto prípade monitorovanie nutnosťou [15].

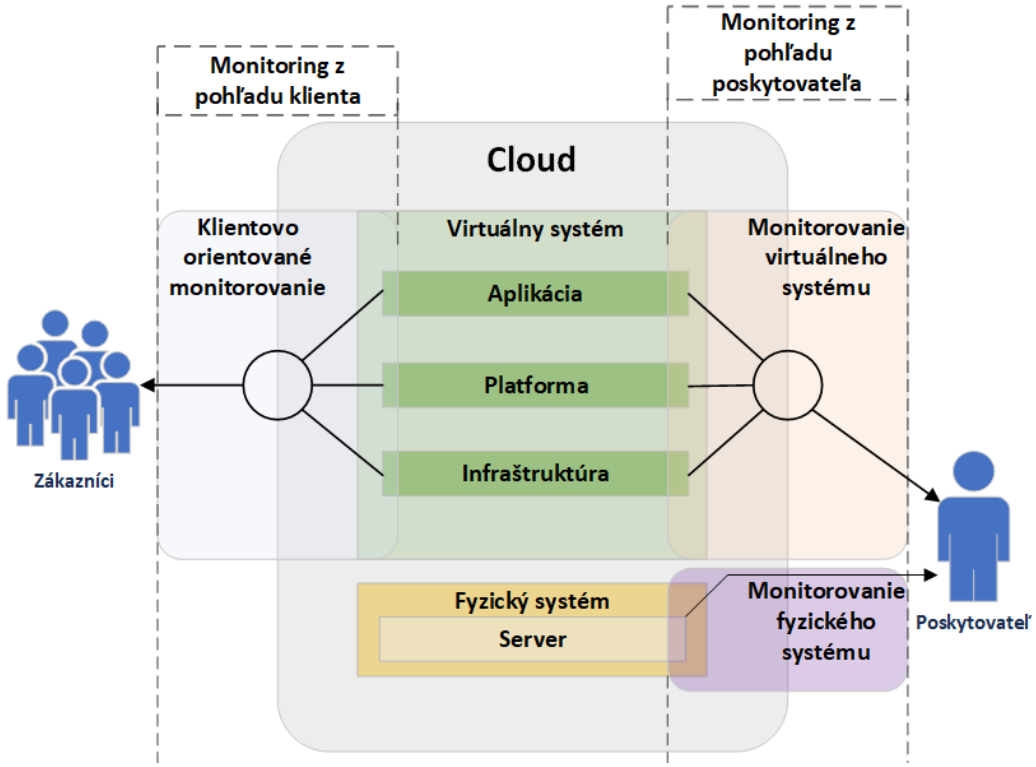
Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
Financované Európskou úniou NextGenerationEU

9.1.8. Bezpečnosť

Bezpečnosť v CC je dôležitá z viacerých dôvodov. V prvom rade je to jeden z hlavných faktorov, ktorý zákazníka odrádza od migrovania jeho infraštruktúry do takéhoto prostredia. Pre manažovanie bezpečnosti v CC infraštruktúre alebo aplikáciách, je potrebný správny monitorovací systém. V prípade poskytovania služieb niektorým štátnym organizáciám musí CC spĺňať prísne kritériá týkajúce sa regulácií ohľadom správy dát. Toto je možné dosiahnuť pomocou monitorovacieho systému, ktorý vie vierohodne dokázať, že používateľove dáta neopustili hranice krajiny [16].

9.2. Pohľady na monitoring

Pre praktické nasadenie riešenia potrebujeme určiť informácie, ktoré bude daný systém zbierať. Primárne sa na problematiku pozeráme z dvoch pohľadov. Popisujeme ich nižšie.



Obrázok 9.1 Pohľady na monitoring [17]

Obrázok 9.1 graficky zobrazuje, čo by sa vo všeobecnosti malo monitorovať a komu dáta z jednotlivých častí prezentovať. Z pohľadu tejto práce nie je zaujímavá časť platformy, ani aplikácie. Môžeme vidieť, že poskytovateľ cloudovej služby (CSP) by mal disponovať informáciami o stave fyzického systému a virtuálnej infraštruktúry, čo v prípade IaaS platformy predstavuje najmä VMs. Zákazník nemá motiváciu sledovať fyzický systém, preto ho zaujíma len stav jeho virtuálnych zdrojov. Poskytovateľ mu môže nejakým spôsobom sprostredkovať takéto informácie.

Keďže model CC je vždy úzko spätý s garanciou kvality služby, užitočné monitorovacie dáta by mali byť späté s podmienkami definovanými v SLA (viď kapitolu **Chyba! Nenašiel sa žiaden zdroj odkazov.**). Definovanie metrík pre obidve formy monitoringu, ako z pohľadu

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
Financované Európskou úniou NextGenerationEU

klienta, tak poskytovateľa, je kritickým aspektom monitorovacieho systému. Exituje však niekoľko problémov, ktoré je potrebné v tomto kontexte brať do úvahy [17].

V prípade klientovo orientovaného monitoringu musia metriky poskytovať informácie pomocou rovnakých parametrov, ako je definované v SLA. Pre príklad uvedieme scenár, kedy CSP garantuje pre každého zákazníka maximálnu kapacitu úložiska pre jeho VM v GigaBajtoch (GB). V takom prípade musí CSP monitorovať úložisko každej VM u všetkých zákazníkov, a následne tieto dáta vhodne interpretovať v rovnakých jednotkách.

V prípade monitorovania virtuálneho systému platí rovnaký princíp. To, čo by sa malo monitorovať, je taktiež úzko späté s SLA. Avšak pre CSP by tieto dáta mali poskytnúť dodatočné informácie [17]. Pre lepšie pochopenie nadviažeme na príklad v prechádzajúcom odseku. Okrem monitorovania úložiska na každej VM by mal CSP zbierať dáta aj o celkovom virtuálnom úložisku v CC platforme. Mal by poznať celkové využitie úložiska všetkými klientami. Tieto informácie by mali slúžiť len pre CSP.

Prípad monitorovania priľahlého fyzického systému je rozdielny od predchádzajúcich dvoch, keďže sa týka nižšej úrovne infraštruktúry, na ktorej stojí celá CC platforma. To znamená, že potrebné metriky sú rovnaké ako pri bežných distribuovaných systémoch. Medzi tieto metriky patria z hardvérovej strany napríklad:

- využitie CPU,
- využitie pamäte RAM,
- využitie fyzického úložiska a
- sieťová prevádzka [17].

Zo strany metrík OS sem môžeme zaradiť napríklad:

- zaťaženie systému a
- využitie virtuálnej pamäte, keďže jej správu má na starosti OS [17].

Takto nazbierané informácie slúžia prioritne pre manažment celého systému. V tomto prípade je spojenie s SLA zakorenené vo vzťahu medzi nízko-úrovňovým fyzickým systémom a vysoko-úrovňovým virtuálnym systémom, ktorý stojí na danej infraštruktúre [17].

Podrobnejšie informácie ku monitorovaniu OpenStack cloudu sú k dispozícií v podpornom dokumente s názvom „Návrh monitorovania OpenStack“.

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
Financované Európskou úniou NextGenerationEU

ZOZNAM ZDROJOV

[1] Techtarget (2025). OpenStack. Available online: https://www.techtarget.com/searchcloudcomputing/definition/OpenStack (accessed 17.2.2025).

[2] L. Chao, Cloud Computing Networking: Theory, Practice, and Development, Boca Raton: CRC Press, 2016.

[3] Denton, J.(2015) Learning OpenStack Networking (Neutron) Second Edition, Birmingham: Packt Publishing Ltd., 2015.

[4] OpenStack (2021), „Openstack Network architecture," 2021. [Online]. Available online: https://docs.openstack.org/arch-design/design-networking.html.

[5] ManageIQ (2024). [Online]. Available online: https://www.manageiq.org/docs/reference/latest/deployment_planning_guide/index.html (accessed 24.3.2025).

[6] COMPUTERWEEKLY. OpenStack Manila: File access storage for the open source cloud Online. 2015 Dostupné z: https://www.computerweekly.com/feature/OpenStack-Manila-File-access-storage-for-the-open-source-cloud. [cit. 2025-04-25].

[7] NASER, Mohammed. An Overview of Cloud File Sharing with OpenStack Manila Online. 2021 Dostupné z: https://vexxhost.com/fr/blog/cloud-file-sharing-with-openstack-manila/. [cit. 2025-04-25].

[8] OPENSTACK WIKI. Manila Online. Dostupné z: https://wiki.openstack.org/wiki/Manila. [cit. 2025-04-25].

[9] OPENSTACK WIKI. Manila/Concepts Online. Dostupné z: https://wiki.openstack.org/wiki/Manila/Concepts. [cit. 2025-04-25].

[10] OPENSTACK WIKI. Glossary Online. 2017 Dostupné z: https://docs.openstack.org/manila/latest/reference/glossary.html#term-manila-api. [cit. 2025-04-25].

[11] RED HAT DOCUMENTATION. Chapter 12. Hardening the Shared File System (Manila) Online. Dostupné z: https://docs.redhat.com/en/documentation/red_hat_openstack_platform/16.1/html/security_and_hardening_guide/hardening_the_shared_file_system_manila. [cit. 2025-04-25].

[12] GITHUB. Charm-manila Online. Dostupné z: https://github.com/openstack/charm-manila/blob/master/TODO.md. [cit. 2025-04-25].

[13] OPENSTACK WIKI. Introduction Online. Dostupné z: https://docs.openstack.org/security-guide/shared-file-systems/intro.html. [cit. 2025-04-25].

[14] C. B. Hauser a S. Wesner, "Reviewing Cloud Monitoring: Towards Cloud Resource Profiling", *IEEE Int. Conf. Cloud Comput. CLOUD*, roč. 2018-July, s. 678–685, 2018, doi: 10.1109/CLOUD.2018.00093.

[15] G. Aceto, A. Botta, W. De Donato, a A. Pescapè, "Cloud monitoring: A survey", *Comput. Networks*, roč. 57, č. 9, s. 2093–2115, 2013, doi: 10.1016/j.comnet.2013.04.001.

[16] L. Romano, D. De Mari, Z. Jerzak, a C. Fetzer, "A novel approach to QoS monitoring in the cloud", *Proc. - 1st Int. Conf. Data Compression, Commun. Process. CCP 2011*, s. 45–51, 2011, doi: 10.1109/CCP.2011.49.

[17] J. Montes, A. Sánchez, B. Memishi, M. S. Pérez, a G. Antoniu, "GMonE: A complete

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
Financované Európskou úniou NextGenerationEU

approach to cloud monitoring", *Futur. Gener. Comput. Syst.*, roč. 29, č. 8, s. 2026–2040, 2013, doi: 10.1016/j.future.2013.02.011.

[18] Web Solutions Inspire Cloud Computing Software | NASA Spinoff. Accessed: Apr. 15, 2024. [Online]. Available: https://spinoff.nasa.gov/Spinoff2012/it_2.html

[19] Who Is Using OpenStack? Accessed: Apr. 15, 2024. [Online]. Available: https://openmetal.io/resources/blog/who-is-using-openstack/

[20] CSNOG 2023 (16-17 May 2023): Distribuovaná NVME storage v hyperkonvergované DC infrastruktuře bez dopadu na bežný síťový provoz. · Indico'. Accessed: Apr. 15, 2024. [Online]. Available: https://indico.csnog.eu/event/13/contributions/117/

[21] openstack.org. 2023. OpenStack Deployment Tools. [online]. 2023. Dostupné na internete: https://www.openstack.org/software/project/navigator/deployment-tools

[22] OpenStack Charmers. 2022. Vault. [online]. 2022. Dostupné na internete: https://charmhub.io/vault

[23] openstack.org. 2022. OpenStack Charms Deployment Guide. [online]. 2022. Dostupné na internete: https://docs.openstack.org/project-deploy-guide/charm-deployment-guide/yoga/index.html

[24] maas.io. 2022. Canonical MAAS. [online]. 2022. Dostupné na internete: https://maas.io/

[25] juju.is. 2022. Take control of [online]. 2022. Dostupné na internete: https://juju.is/

[26] T. G. Peter Mell, „The NIST Definition of Cloud," The National Institute of Standards and Technology, September 2011. [Online]. Available: http://faculty.winthrop.edu/domanm/csci411/Handouts/NIST.pdf. [Cit. 13 01 2020].

[27] BMC SOFTWARE, INC. Distributed Storage Model - Concept and Principles - Documentation for BMC Discovery content reference - BMC Documentation [online]. [Dátum 29.3.2024]. Dostupné na internete: https://docs.bmc.com/docs/discovery/contentref/distributed-storage-model-concept-and-principles-997880678.html

[28] NUTANIX. What is Distributed Storage? Explore Distributed Cloud | Nutanix [online]. [Dátum 29.3.2024]. Dostupné na internete: https://www.nutanix.com/info/distributed-storage

[29] TELNYX. What is distributed storage, and why does it matter? [online]. [Dátum 29.3.2024]. Dostupné na internete: https://telnyx.com/resources/what-is-distributed-storage

[30] WEKA.IO. Distributed File Systems Explained | Features and Advantages [online]. [Dátum 29.3.2024]. Dostupné na internete: https://www.weka.io/learn/distributed-file-systems/distributed-file-system/

[31] HOSTKEY. What is Distributed Storage? Types and Examples [online]. [Dátum 21.3.2024]. Dostupné na internete: https://hostkey.com/blog/53-what-is-distributed-storage/

[32] CLOUDIAN, INC. Distributed Storage: What's Inside Amazon S3? [online]. [Dátum 21.3.2024]. Dostupné na internete: https://cloudian.com/guides/data-backup/distributed-storage/

[33] JULIO'S BLOG. Selecting the right storage backend for your OpenStack cloud [online]. [Dátum 9.10.2023]. Dostupné na internete: https://www.juliosblog.com/selecting-the-right-storage-backend-for-your-openstack-cloud/

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
Financované Európskou úniou NextGenerationEU

[34] STORAGE SWISS. 2015. Selecting the right OpenStack storage module [online]. [Dátum 9.10.2023]. Dostupné na internete: https://storageswiss.com/2015/10/14/selecting-the-right-openstack-storage-module/

[35] CEPH DOCUMENTATION. Reef [online]. [Dátum 9.10.2023]. Dostupné na internete: https://docs.ceph.com/en/reef/

[36] OPENSTACK. 2023. OpenStack Administration Guide [online]. [Dátum 9.10.2023]. Dostupné na internete: https://docs.openstack.org/2023.2/admin/

[37] GLUSTER DOCS. Latest documentation [online]. [Dátum 19.10.2023]. Dostupné na internete: https://docs.gluster.org/en/latest/

[38] RED HAT. 3.3. Red Hat Gluster Storage 3.3 Administration Guide [online]. [Dátum 19.10.2023]. Dostupné na internete: https://access.redhat.com/documentation/en-us/red_hat_gluster_storage/3.3/html-single/administration_guide/index#sect-Managing_Block_Storage

[39] MINIO. MinIO [online]. [Dátum 19.10.2023]. Dostupné na internete: https://min.io/

[40] CEPH. Architecture — Ceph Documentation [online]. [Dátum 15.2.2024]. Dostupné na internete: https://docs.ceph.com/en/reef/architecture/

[41] CEPH FOUNDATION. Ceph.io — About the foundation [online]. [Dátum 21.3.2024]. Dostupné na internete: https://ceph.io/en/foundation/about/

[42] CEPH. Red Hat's Ceph team is moving to IBM [online]. [Dátum 21.3.2024]. Dostupné na internete: https://ceph.io/en/news/blog/2022/red-hats-ceph-team-is-moving-to-ibm/

[43] CEPH AUTHORS AND CONTRIBUTORS. Ceph Releases (general) — Ceph Documentation [online]. [Dátum 21.3.2024]. Dostupné na internete: https://docs.ceph.com/en/latest/releases/general/

[44] HG INSIGHTS. Companies Using Ceph, Market Share, Customers and Competitors [online]. [Dátum 11.3.2024]. Dostupné na internete: https://discovery.hgdata.com/product/ceph

[45] Canonical Ltd. What is Ceph? | Ubuntu [online]. [Dátum 18.2.2024]. Dostupné na internete: https://ubuntu.com/ceph/what-is-ceph

[46] CEPH AUTHORS AND CONTRIBUTORS. Intro to Ceph — Ceph Documentation [online]. [Dátum 15.2.2024]. Dostupné na internete: https://docs.ceph.com/en/latest/start/intro/

[47] CEPH. Monitor Config Reference — Ceph Documentation [online]. [Dátum 15.2.2024]. Dostupné na internete: https://docs.ceph.com/en/reef/rados/configuration/mon-config-ref/

[48] CEPH. Ceph Manager Daemon — Ceph Documentation [online]. [Dátum 15.2.2024]. Dostupné na internete: https://docs.ceph.com/en/reef/mgr/

[49] CEPH. OSD Service — Ceph Documentation [online]. [Dátum 21.2.2024]. Dostupné na internete: https://docs.ceph.com/en/reef/cephadm/services/osd/

[50] CICIMOV, Igor. Ceph cluster on Ubuntu-14.04 - DevOps [online]. [Dátum 21.1.2024]. Dostupné na internete: https://icicimov.github.io/blog/storage/Ceph-cluster-on-Ubuntu-14.04/

[51] CEPH. ceph-mds -- ceph metadata server daemon — Ceph Documentation [online]. [Dátum 16.2.2024]. Dostupné na internete: https://docs.ceph.com/en/reef/man/8/ceph-mds/

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
Financované Európskou úniou NextGenerationEU

[52] CEPH. MDS Journaling — Ceph Documentation [online]. [Dátum 21.2.2024]. Dostupné na internete: https://docs.ceph.com/en/reef/cephfs/mds-journaling/

[53] WEIL, Sage, LEUNG, Andrew, BRANDT, Scott, MALTZAHN, Carlos, 2007. RADOS: A scalable, reliable storage service for petabyte-scale storage clusters. In: Proceedings of the 2nd International Workshop on Petascale Data Storage. pp. 35-44. [Dátum 22.2.2024]. DOI: 10.1145/1374596.1374606.

[54] FLYDEAN. 04 Multi Paxos [online]. [Dátum 22.2.2024]. Dostupné na internete: https://docs.flydean.com/www.flydean.com/distribution/04-multi-paxos

[55] CEPH AUTHORS AND CONTRIBUTORS. Ceph Storage Cluster — Ceph Documentation [online]. [Dátum 22.2.2024]. Dostupné na internete: https://docs.ceph.com/en/reef/rados/

[56] CHUM, Sopanhapich, PARK, Heekwon and CHOI, Jongmoo, 2021. Supporting SLA via Adaptive Mapping and Heterogeneous Storage Devices in Ceph. Electronics. Vol. 10, p. 847. [Dátum 22.2.2024]. DOI: 10.3390/electronics10070847.

[57] CEPH AUTHORS AND CONTRIBUTORS. Introduction to librados — Ceph Documentation [online]. [Dátum 23.2.2024]. Dostupné na internete: https://docs.ceph.com/en/latest/rados/api/librados-intro/

[58] CEPH AUTHORS AND CONTRIBUTORS. CRUSH Maps — Ceph Documentation [online]. [Dátum 19.2.2024]. Dostupné na internete: https://docs.ceph.com/en/latest/rados/operations/crush-map/

[59] CEPH AUTHORS AND CONTRIBUTORS. Pools — Ceph Documentation [online]. [Dátum 13.3.2024]. Dostupné na internete: https://docs.ceph.com/en/latest/rados/operations/pools/

[60] CEPH AUTHORS AND CONTRIBUTORS. Placement Groups — Ceph Documentation [online]. [Dátum 14.3.2024]. Dostupné na internete: https://docs.ceph.com/en/reef/rados/operations/placement-groups/

[61] D'ATRI, Anthony, Vaibhav BHAMBRE, and Karan SINGH. Learning Ceph - Second Edition. Birmingham: Packt Publishing, 2017. ISBN 978-1787127913.

[62] CEPH AUTHORS AND CONTRIBUTORS. PG (Placement Group) notes — Ceph Documentation [online]. [Dátum 14.3.2024]. Dostupné na internete: https://docs.ceph.com/en/latest/dev/placement-group/

[63] „Newton install everview“ [Online] [cit. Marec 2021]. Avaible:https://docs.openstack.org/newton/install-guide-rdo/overview.html, 2019.

[64] Anthony Critelli 2020. „Using Keepalived for managing simple failover in clusters“ [Online] [cit. Apríl 2021]. Avaible: https://www.redhat.com/sysadmin/ha-cluster-linux

[65] “33+ Cloud Computing Statistics, Facts, and Trends [2020 Updated].” https://techjury.net/blog/cloud-computing-statistics/#gref (accessed Mar. 16, 2021).

[66] “Rackspace Technology | Multicloud Solutions Provider.” https://www.rackspace.com/ (accessed Apr. 13, 2021).

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
Financované Európskou úniou NextGenerationEU