Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
(повне найменування вищого навчального закладу)
Факультет комп'ютерно-інформаційних систем і програмної інженерії
(назва факультету)
Кафедра комп'ютерних систем та мереж
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр
(освітній рівень)

на тему: Методи та засоби віртуалізації для створення високодоступних комп'ютеризованих систем

Виконав: студент (ка) VI курсу, групи СІм-62
Спеціальності:
123 "Комп'ютерна інженерія"
(шифр і назва спеціальності)

Чех Т. П.
підпис (прізвище та ініціали)

Керівник
Луцик Н. С.
підпис (прізвище та ініціали)

Нормоконтроль
Тиш Є.В.
підпис (прізвище та ініціали)

Завідувач кафедри
Осухівська Г. М.
підпис (прізвище та ініціали)

Рецензент
Муж В.В.
підпис (прізвище та ініціали)

м. Тернопіль – 2023

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)
Кафедра комп'ютерних систем та мереж
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ *Осухівська Г. М.*
(підпис) (прізвище та ініціали)
«__» _____ 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня *магістр*
(назва освітнього ступеня)
за спеціальністю *123 Комп'ютерна інженерія*
(шифр і назва спеціальності)
Студенту *Чеху Тарасу Павловичу*
(прізвище, ім'я, по батькові)
1. Тема роботи *Методи та засоби віртуалізації для створення високодоступних комп'ютеризованих систем*

Керівник роботи *Луцик Надія Степанівна*
доктор філософії, доцент кафедри комп'ютерних систем та мереж
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
Затверджені наказом ректора *від « 01 » 12 2023 року № 4/7-1132*
2. Термін подання студентом завершеної роботи *22.12.2023*
3. Вихідні дані до роботи *Вимоги до високодоступних систем, операційна система Oracle Linux, NAS сревер.*

4. Зміст роботи (перелік питань, які потрібно розробити)
Вступ.
1. Аналіз рішень для створення високодоступних систем з використанням кластерів віртуалізації.
2. Принципи та особливості роботи гіпервізора KVM.
3. Розробка кластера високої доступності на базі Oracle Linux.
4. Охорона праці та безпека в надзвичайних ситуаціях.
Висновки.
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)
1. Актуальність теми, мета і завдання дослідження, об'єкт та предмет дослідження.
2. Методи дослідження, наукова новизна та практичні результати.
3. Типи гіпервізорів. 4. Кластери віртуалізації.
5. Принципи та особливості роботи гіпервізора KVM.
6. Архітектура Oracle Linux Virtualization Manager.
7. Взаємодія компонентів oVirt з libvirt, VDSM, QEMU та KVM в Oracle Linux.
8. Схема тестового середовища для створення кластера високої доступності.
9. Блок-схема алгоритму роботи кластера високої доступності.
10. Висновки.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	*Осухівська Г.М., к.т.н., доцент*		
Безпека в надзвичайних ситуаціях	*Клепчик В.М., проректор з адміністративно-господарської роботи та будівництва*		

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	*Вступ*		*Виконано*
2.	*Аналіз рішень для створення високодоступних систем з використанням кластерів віртуалізації.*		*Виконано*
3.	*Принципи та особливості роботи гіпервізора KVM.*		*Виконано*
4.	*Розробка кластера високої доступності на базі Oracle Linux.*		*Виконано*
5.	*Охорона праці та безпека в надзвичайних ситуаціях*		*Виконано*
6.	*Висновки.*		*Виконано*
7.	*Оформлення пояснювальної записки*		*Виконано*
8.	*Оформлення графічної частини*		*Виконано*
9.	*Попередній захист кваліфікаційної роботи магістра*		*Виконано*
10.	*Захист кваліфікаційної роботи*		

Студент _____ *Чех Т.П.*
(підпис) (прізвище та ініціали)

Керівник роботи _____ *Луцик Н.С.*
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Методи та засоби віртуалізації для створення високодоступних комп’ютеризованих систем //Кваліфікаційна робота магістра // Чех Тарас Павлович// Тернопільський національний технічний університет імені Івана Пулюя, факультет комп’ютерно-інформаційних систем і програмної інженерії, кафедра комп’ютерних систем та мереж, група СІм-62 // Тернопіль, 2023 // с. – 98, рис. – 38, табл. – 1, аркушів А1 – 10, додат. – 2, бібліогр. – 19.

Ключові слова: кластер, віртуалізація, KVM, Oracle, надійність, QEMU, Linux.

Кваліфікаційну роботу магістра присвячено дослідженню технології віртуалізації в кластерах високої доступності. Проведено огляд та порівняння сучасних методів створення кластерів на основі платформ віртуалізації VMware ESXi, Microsoft Hyper-V, Citrix Hypervisor та Oracle Linux KVM. Досліджено їхні можливості у забезпеченні високої доступності системи у випадку відмови одного чи кількох хостів кластера. Розглянуто принципи та особливості роботи гіпервізора KVM та QEMU. Також розглянуто використання гіпервізором KVM апаратної віртуалізації Intel VT-x та AMD-V для оптимізації роботи віртуальних машин. Проведено опис архітектури Oracle Linux Virtualization Manager та створено схему взаємодії компонентів кластера, що базується на хостах Oracle Linux KVM та мережевому сховищі даних TrueNAS CORE. Створено кластер високої доступності та проведено процес тестування, спрямований на перевірку надійності та стійкості системи. Проведено оцінку здатності системи ефективно керувати віртуальними ресурсами та забезпечувати стабільну роботу під час проблем з окремими хостами кластера.

ABSTRACT

Methods and tools for virtualization to create highly accessible computerized systems// Master's graduation thesis// Chekh Taras // Ivan Pulyu Ternopil National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Computer Systems and Networks, group СІм-62 // Ternopil, 2023 // p. – 98, fig. - 38, tab. - 1, sheets A1 - 10, add. – 2, bibliography - 19.

Keywords: cluster, virtualization, KVM, Oracle, reliability, QEMU, Linux.

The master's thesis is devoted to the research of virtualization technology in high-availability clusters. A review and comparison of modern methods of creating clusters based on virtualization platforms VMware ESXi, Microsoft Hyper-V, Citrix Hypervisor and Oracle Linux KVM was carried out. Their capabilities in ensuring high availability of the system in case of failure of one or more cluster hosts were studied. The principles and features of the KVM and QEMU hypervisor are considered. The use of Intel VT-x and AMD-V hardware virtualization by the KVM hypervisor to optimize the operation of virtual machines is also considered. The architecture of Oracle Linux Virtualization Manager has been described and the interaction diagram of cluster components based on Oracle Linux KVM hosts and TrueNAS CORE network data storage has been created. A high-availability cluster was created and a testing process aimed at checking the reliability and stability of the system was carried out. The system's ability to effectively manage virtual resources and ensure stable operation during problems with individual cluster hosts was evaluated.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,СКОРОЧЕНЬ І ТЕРМІНІВ 8
ВСТУП..... 9
РОЗДІЛ 1 АНАЛІЗ РІШЕНЬ ДЛЯ СТВОРЕННЯ ВИСОКОДОСТУПНИХ СИСТЕМ З ВИКОРИСТАННЯМ КЛАСТЕРІВ ВІРТУАЛІЗАЦІЇ..... 11
1.1. Огляд технології віртуалізації..... 11
1.2. Огляд кластерів віртуалізації 14
1.3. Огляд рішення віртуалізації від VMware 16
1.4. Огляд Hyper-V від Microsoft 21
1.5. Огляд технології віртуалізації XEN від Citrix 24
1.6. Огляд реалізації KVM від Oracle 25
1.7. Порівняння кластерів віртуалізації 27
1.8. Висновки до розділу 29
РОЗДІЛ 2 ПРИНЦИПИ ТА ОСОБЛИВОСТІ РОБОТИ ГІПЕРВІЗОРА KVM ... 30
2.1. Принцип роботи гіпервізора KVM..... 30
2.2. Апаратна віртуалізація Intel VT-x та AMD-V 33
2.3. Взаємодія KVM та QEMU 40
2.4. Перевірка підтримки віртуалізації 44
2.5. Висновки до розділу 47
РОЗДІЛ 3 РОЗРОБКА КЛАСТЕРА ВИСОКОЇ ДОСТУПНОСТІ НА БАЗІ ORACLE LINUX..... 48
3.1. Архітектура Oracle Linux Virtualization Manager 48
3.2. Планування та розгортання кластера високої доступності 55
3.2.1. Центр обробки даних 56
3.2.2. Кластер 57
3.2.3. Хости Oracle Linux KVM..... 58
3.2.4. Віртуальні машини..... 59
3.2.5. Конфігурація мережі 60
3.2.6. Засоби зберігання 64

3.2.7. Висока доступність і оптимізація 70
3.3. Тестування кластера високої доступності 75
3.4. Висновки до розділу 84
РОЗДІЛ 4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ 85
4.1. Охорона праці 85
4.2. Безпека в надзвичайних ситуаціях 88
4.2.1. Державна система моніторингу довкілля, як складова частина національної інформаційної інфраструктури, сумісної з аналогічними системами інших країн 88
4.2.2. Оцінка стійкості роботи промислового підприємства до дії світлового випромінювання ядерного вибуху 92
4.3. Висновки до розділу 95
ВИСНОВКИ 96
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 97
ДОДАТКИ 99
Додаток А Тези конференцій 99
Додаток Б Блок-схема алгоритму роботи кластера 107

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

VMM	Virtual Machine Monitor
ВМ	Віртуальна машина
KVM	Kernel-based Virtual Machine
ОС	Операційна система
ПЗ	Програмне забезпечення
DRS	Distributed Resource Scheduler
HA	High Availability
KVM	Kernel-based Virtual Machine
OLVM	Oracle Linux Virtualization Manager
NFS	Network File System
VMX	Virtual Machine Extensions
EPT	Extended Page Tables
VMCS	Virtual Machine Control Structure
SVM	Secure Virtual Machine
QEMU	Quick Emulator
TCG	Tiny Code Generator
VDSM	Virtual Desktop and Server Manager
VNC	Virtual Network Computing
iSCSI	Internet Small Computer System Interface
NAS	Network Attached Storage
SAN	Storage Area Network
SPM	Storage Pool Manager
MoM	Memory Overcommitment Manager
KSM	Kernel Same-page Merging
API	Application Programming Interface

ВСТУП

Актуальність теми. Швидкий розвиток технологій обробки даних та інформаційних систем потребує забезпечення стійкості, надійності та безперервності роботи систем. Забезпечення безперервності роботи інформаційних сервісів набуває все більшої важливості для підприємств у зв'язку зі зростанням обсягів даних та їх критичності для бізнесу. Ця проблема актуальна для широкого спектру галузей та вимагає системного підходу до розробки та впровадження високодоступних інформаційних систем. У цьому контексті, створення високодоступних систем за допомогою методів та засобів віртуалізації є надзвичайно актуальною темою.

Мета і завдання дослідження. Метою даного дослідження є аналіз та порівняння кластерів віртуалізації на основі Hyper-V, VMware ESXi, XEN та KVM з подальшим створення високодоступних систем з використанням кластерів віртуалізації на основі KVM.

Основні завдання дослідження:

- аналізу основних платформ віртуалізації;
- аналіз можливостей гіпервізора KVM
- розробка концепції створення високодоступних систем на базі KVM;
- розробка кластера високої доступності на базі KVM;
- тестування та оцінка результатів.

Об’єкт дослідження. Об’єктом дослідження є кластери віртуалізації та їхні можливості для створення високодоступних систем.

Предмет дослідження. Предметом дослідження є методи та засоби віртуалізації, зокрема аналіз функціоналу та характеристик платформ Hyper-V, ESXi, XEN та KVM для створення високодоступних систем. Окрім того, предметом дослідження є можливість створення кластера високої доступності на базі KVM з використанням TrueNAS CORE сервера.

Методи дослідження. Для досягнення мети й вирішення завдань застосовувалися різноманітні методи наукового дослідження.

Такі як:

- аналіз та порівняння варіантів рішень;

- експериментальні дослідження з проведенням тестів кластера віртуалізації;

- моделювання аварійних ситуацій для перевірки функціональності високої доступності.

Наукова новизна одержаних результатів кваліфікаційної роботи. Наукова новизна у кваліфікаційній роботі полягає в:

- унікальному поєднанні досліджень та практичних застосувань у галузі віртуалізації для створення високодоступних систем;

- вдосконаленні методу створення високодоступних систем з використанням платформи віртуалізації KVM в поєднанні з TrueNAS CORE сервером.

Практичне значення одержаних результатів. Отримані результати можуть бути використані у сфері проектування, розробки та управління інформаційними системами для підвищення їх доступності, надійності та ефективності.

Публікації. Результати дослідження апробовано на XI науково-технічної конференції «Інформаційні моделі, системи та технології», V міжнародна науково-практична конференція «Scientific practice: modern and classical research methods».

Структура роботи. Робота складається з пояснювальної записки та графічної частини. Пояснювальна записка складається із вступу, 4 розділів, висновків, списку використаних джерел та додатку (-ів). Обсяг роботи: пояснювальна записка – 98 аркушів формату А4, графічна частина – 10 аркушів формату А1.

РОЗДІЛ 1
АНАЛІЗ РІШЕНЬ ДЛЯ СТВОРЕННЯ ВИСОКОДОСТУПНИХ СИСТЕМ З ВИКОРИСТАННЯМ КЛАСТЕРІВ ВІРТУАЛІЗАЦІЇ

1.1. Огляд технології віртуалізації

Віртуалізація - це процес створення віртуальних (абстрагованих) версій ресурсів апаратного забезпечення, таких як обчислювальна потужність, зберігання даних, мережеві ресурси і т.д. Це дозволяє використовувати ці ресурси більш ефективно, розділяючи їх між декількома віртуальними середовищами.

Основні поняття в віртуалізації включають: гіпервізори, віртуальні машини, ізоляцію ресурсів, міграцію та резервне копіювання, масштабованість, безпеку.

Гіпервізор (Hypervisor) або VMM - це ключовий компонент технології віртуалізації, який дозволяє створювати та управляти віртуальними машинами [1]. Гіпервізори розділяють фізичний сервер на окремі віртуальні середовища, кожне з яких може мати свою операційну систему та власний набір ресурсів.

Існують два типи гіпервізорів.

Тип 1 (native, bare-metal) - цей тип гіпервізора працює безпосередньо на апаратному рівні, напряму контролюючи апаратні ресурси сервера (див.рис.1.1).

Він встановлюється безпосередньо на фізичний сервер як операційна система і управляє ресурсами для віртуальних машин. Приклади таких гіпервізорів: VMware ESXi, Microsoft Hyper-V Server, XEN а також KVM.

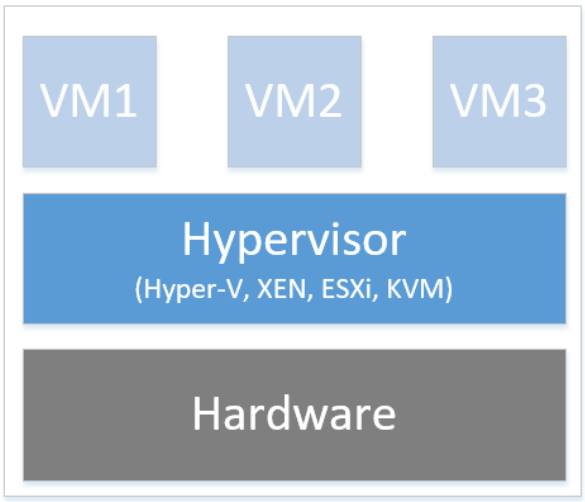


Рис. 1.1. Гіпервізор тип 1

Тип 2 (hosted) - цей тип гіпервізора працює поверх операційної системи як звичайна програма (див.рис.1.2).

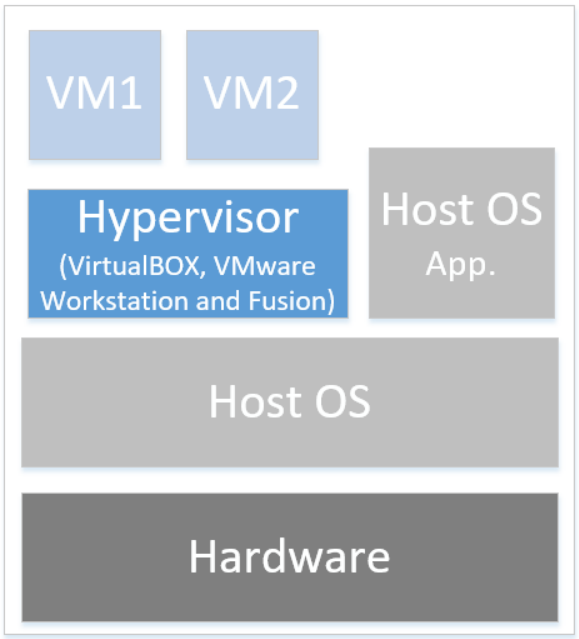


Рис. 1.2. Гіпервізор тип 2

Користувач спочатку запускає операційну систему на фізичному сервері, а потім встановлює гіпервізор як додаток, який управляє віртуальними машинами. Приклади таких гіпервізорів: Oracle VirtualBox, VMware Workstation, Parallels Desktop.

Гіпервізор дозволяє створювати, запускати, зупиняти та видаляти віртуальні машини. Він виділяє апаратні ресурси фізичного сервера для використання в різних віртуальних середовищах. Віртуальні машини, які керуються гіпервізором, є ізольованими одна від одної, що забезпечує безпеку та надійність системи. Гіпервізори надають можливість створення резервних копій віртуальних машин, а також переміщення їх між фізичними серверами без зупинки роботи.

Гіпервізори є основними елементами віртуалізації, оскільки вони дозволяють оптимально використовувати ресурси сервера та забезпечують ізоляцію різних робочих середовищ. Обраний тип гіпервізора може вплинути на продуктивність, безпеку та можливості управління віртуальною інфраструктурою.

Віртуальна машина є ізольованим середовищем, що відтворює функціональність фізичного комп'ютера, проте працює в області програмного забезпечення. ВМ має власну ОС, додатки, пам'ять, процесор та ресурси зберігання, тобто усі компоненти, необхідні для функціонування окремого комп'ютера.

ВМ ізольовані одна від одної та від хост-системи. Це означає, що вони працюють у власному віртуальному середовищі, не взаємодіючи напряму одна з одною. Кожна віртуальна машина може мати свою власну ОС. Це дозволяє запускати різні ОС (наприклад, Windows, Linux, MacOS) на одному фізичному сервері.

ВМ отримують віртуальні ресурси, такі як процесорний час, оперативна пам'ять та диск, які відділяються від реальних фізичних ресурсів. Також є можливість зміни ресурсів, виділених ВМ, в залежності від потреби. Наприклад, збільшення обсягу оперативної пам'яті або кількості процесорних ядер.

Є можливість створювати шаблони для швидкого розгортання нових ВМ та створення зрізів стану ВМ у певний момент часу для подальшого відновлення.

ВМ можуть бути переміщені між різними серверами або хостами без втрати продуктивності, що дозволяє оптимізувати ресурси та уникнути відмов.

Віртуальні машини ізольовані від інших, що підвищує рівень безпеки. Кожна ВМ може мати власні правила безпеки та доступу до ресурсів.

Віртуальні машини використовуються для багатьох цілей, включаючи тестування програмного забезпечення, розробку, вебхостинг, консолідацію серверів та управління ресурсами обчислювальних систем. Вони дозволяють ефективно використовувати обчислювальні ресурси інфраструктури.

Віртуалізація дозволяє оптимізувати використання апаратного забезпечення, підвищує гнучкість та надійність систем, спрощує процеси управління та підтримки інфраструктури. Вона є ключовою технологією для багатьох сучасних інформаційних та комп'ютерних систем, що дозволяє забезпечити більш ефективне та оптимізоване використання обчислювальних ресурсів.

1.2. Огляд кластерів віртуалізації

Кластер віртуалізації - це група фізичних серверів, які об'єднуються для створення великого пула ресурсів, які можуть бути використані для віртуальних машин або інших віртуалізованих середовищ. Ця технологія дозволяє забезпечити високу доступність, балансування навантаження та толерантність до відмов, забезпечуючи продовження роботи навіть у випадку відмови одного або декількох серверів.

Кластер віртуалізації дозволяє створювати системи, які можуть продовжувати роботу в разі відмови одного або декількох серверів.

Висока доступність (high availability) - це характеристика системи, яка означає здатність інфраструктури або програмного забезпечення продовжувати працювати без перебоїв та забезпечувати доступ до сервісів навіть після відмови окремих компонентів [2]. Вона стає критично важливою для бізнесу, що вимагає безперервної роботи систем та служб.

Система має мати механізми резервного копіювання даних та можливість швидкого відновлення з них у випадку відмови апаратного забезпечення чи

програмного збою. А також механізм дублювання чи реплікації даних та служб на кількох серверах чи вузлах для забезпечення доступності навіть при відмові одного з серверів.

Висока доступність важлива для багатьох галузей, таких як фінанси, медицина, інформаційні технології та інші, де навіть невеликі перерви у роботі можуть призвести до значних фінансових втрат чи проблем для користувачів. Використання систем з високою доступністю є критично важливим для забезпечення стабільної та безперервної роботи.

Кластер може розподіляти робочі завдання та навантаження між різними серверами для оптимізації використання ресурсів та підвищення продуктивності.

Балансування навантаження (load balancing) - це метод розподілу робочих завдань та запитів між різними ресурсами (серверами, комп'ютерами, мережевими з'єднаннями тощо) у системі, з метою оптимізації використання ресурсів та забезпечення ефективності роботи.

Балансування навантаження дозволяє оптимізувати використання ресурсів та забезпечити оптимальну продуктивність. Наприклад, використання менше завантажених серверів для виконання завдань, тим самим запобігаючи перевантаженню.

Розподіл робочих завдань між різними ресурсами також може забезпечити високу доступність. Якщо один сервер виявляється недоступним, балансування навантаження може автоматично перенести робочі завдання на інший доступний ресурс [3].

Система балансування навантаження може бути налаштована для відповіді на збільшення навантаження шляхом динамічного розподілу завдань на додаткові ресурси.

Балансування навантаження є важливим моментом при створенні високодоступних, ефективних та надійних інфраструктур. Це дозволяє раціонально використовувати ресурси, забезпечуючи максимальну продуктивність та доступність для користувачів.

Кластер може бути розширений додаванням нових серверів або вузлів для збільшення обсягу ресурсів або підтримки зростаючого навантаження.

Багато систем кластерної віртуалізації надають централізовані інструменти управління, які дозволяють керувати всім кластером через єдиний інтерфейс.

Популярні технології кластерів віртуалізації включають VMware vSphere, Microsoft Hyper-V Failover Cluster, Red Hat Virtualization Manager, Oracle Linux Virtualization Manager, Xen Orchestra.

Технологія кластерів віртуалізації є ключовою для створення надійних, ефективних та високодоступних інфраструктур віртуалізації, що дозволяє забезпечити стійкість до відмов та ефективне використання ресурсів.

1.3. Огляд рішення віртуалізації від VMware

VMware - один із провідних постачальників рішень у сфері віртуалізації. Компанія пропонує широкий спектр продуктів і рішень для віртуалізації серверів, робочих станцій, мереж та хмарних інфраструктур.

VMware vSphere - це комплексна платформа для віртуалізації і управління обчислювальними, мережевими та іншими ресурсами в iT-інфраструктурі [4]. Ця платформа включає в себе кілька ключових компонентів, які працюють разом для створення та управління віртуальними середовищами: VMware ESXi, vCenter Server, vSphere Web Client, vSphere Distributed Resource Scheduler, vSphere High Availability, vSphere vMotion, vSphere Fault Tolerance.

VMware ESXi є гіпервізором типу 1, що працює на рівні апаратного забезпечення сервера, безпосередньо контролюючи доступ до ресурсів та відповідаючи за управління віртуальними машинами.

ESXi працює безпосередньо на рівні апаратного забезпечення та не потребує операційної системи для функціонування. Це дозволяє забезпечити високу продуктивність та ефективність віртуалізації, оскільки гіпервізор працює без додаткових шарів ПЗ.

Гіпервізор надає віртуальним машинам доступ до фізичних ресурсів сервера, таких як процесори, пам'ять, мережа та ресурси зберігання, і керує цими ресурсами для ефективного використання.

Інтерфейс користувача ESXi надає можливість адміністраторам легко налаштовувати, моніторити та керувати віртуальними машинами та ресурсами сервера.

ESXi має різні функції для захисту віртуальних середовищ, включаючи механізми для ізоляції віртуальних машин одна від одної та механізми захисту доступу до системи.

VMware vCenter Server є центральним елементом у платформі VMware vSphere, призначеним для централізованого управління та моніторингу віртуальної інфраструктури. Він забезпечує адміністраторам централізований доступ до всіх компонентів віртуалізації VMware, дозволяючи ефективно керувати та налаштовувати інфраструктуру в одному місці. vCenter Server дозволяє адміністраторам створювати, запускати, зупиняти, переміщувати та видаляти ВМ на різних серверах ESXi. Адміністратори можуть ефективно розподіляти ресурси (процесори, пам'ять, зберігання) між віртуальними машинами. vCenter Server надає можливість моніторингу використання ресурсів та виконання міграції ВМ з одного сервера ESXi на інший.

Платформа надає можливості для управління доступом користувачів, створення ролей, аудиту подій та застосування політик безпеки для віртуальних середовищ. Забезпечує можливість створення резервних копій віртуальних машин та відновлення їх у випадку відмови або аварій.

Інтерфейс vSphere Web Client є вебінтерфейсом користувача для управління віртуальною інфраструктурою VMware vSphere. Цей інтерфейс дозволяє адміністраторам працювати з різними компонентами vSphere через веббраузер, надаючи розширений функціонал.

За допомогою вебінтерфейсу адміністратори можуть керувати різними аспектами віртуальної інфраструктури, такими як віртуальні машини, хост-системи, мережі та зберігання, з одного централізованого місця. vSphere Web

Client має сучасний та інтуїтивно зрозумілий інтерфейс, що полегшує роботу з віртуальною інфраструктурою та дозволяє швидко здійснювати різні операції.

Він надає доступ до різних функцій, таких як управління міграцією ВМ, розподіл ресурсів , висока доступність, створення резервних копій, моніторинг та ряд інших.

За допомогою вебклієнта декілька адміністраторів можуть спільно працювати над управлінням віртуальними ресурсами. Вебклієнт надає доступ до розширених можливостей налаштування і адміністрування для компонентів vSphere, дозволяючи конфігурувати та керувати різними параметрами системи.

Завдяки vSphere Web Client адміністратори можуть ефективно керувати і моніторити віртуальну інфраструктуру VMware з будь-якого місця, де є доступ до мережі, що робить його потужним інструментом для управління інфраструктурою віртуалізації.

VMware vSphere Distributed Resource Scheduler - це функціонал, що надає автоматизоване розподілення та оптимізацію ресурсів ВМ на рівні кластера серверів виртуалізації vSphere.

DRS аналізує використання ресурсів (такі як процесор, пам'ять, зберігання) серед віртуальних машин на серверах у кластері і автоматично розподіляє ці ресурси для оптимізації продуктивності та уникнення перевантаження чи нестачі ресурсів. DRS може автоматично переміщувати ВМ між фізичними серверами в кластері для забезпечення більш рівномірного навантаження між серверами, підвищуючи продуктивність та розподіл ресурсів. При зміні навантаження DRS може адаптувати конфігурацію ресурсів в кластері шляхом автоматичного розподілу навантаження між серверами. Адміністратори можуть налаштовувати політики DRS для визначення рівня автоматизації та чутливості до змін, а також для визначення того, коли і як часто DRS повинен втручатися у розподіл ресурсів. DRS може сприяти підвищенню доступності системи шляхом розміщення ВМ у відповідних серверах з урахуванням рівня навантаження та запобігання збільшенню одномоментного навантаження на окремі ресурси.

Використання DRS дозволяє оптимізувати використання ресурсів, поліпшує продуктивність віртуальної інфраструктури, забезпечуючи балансування ресурсів та надійність системи у середовищі віртуалізації VMware vSphere.

VMware vSphere High Availability - це функціонал, призначений для автоматичного відновлення роботи ВМ у випадку відмови апаратного забезпечення або програмних збоїв на рівні сервера.

HA надає механізм для автоматичного перезапуску ВМ на доступних серверах у випадку відмови апаратного забезпечення, таким чином, забезпечуючи безперервну роботу додатків та сервісів. Механізм постійно моніторить стан фізичних серверів у кластері. Якщо сервер виявляється недоступним через відмову апаратного забезпечення або іншу причину, HA автоматично перезапускає ВМ на інших доступних серверах.

Адміністратори можуть налаштовувати різні параметри HA, такі як пріоритет перезапуску VM, конфігурацію мережі для доступу до відновлених ВМ та інші опції. HA допомагає забезпечити високу доступність середовища віртуалізації, уникнення простоїв у роботі системи та недоступності сервісів.

VMware vMotion - це функціонал, який дозволяє переміщення в реальному часі (live migration) віртуальних машин між фізичними серверами у віртуальній інфраструктурі VMware vSphere без зупинки роботи ВМ або перерви в доступі до послуг.

Механізм дозволяє адміністраторам переміщувати робочі ВМ між різними фізичними серверами у віртуальному кластері без будь-якої перерви в роботі або втрати доступу до сервісів, що працюють на ВМ. vMotion дозволяє розподіляти ресурси між фізичними серверами у віртуальному кластері, оптимізуючи використання обчислювальних ресурсів та забезпечуючи балансування навантаження. Користувачі можуть продовжувати користуватися послугами, навіть під час процесу переміщення ВМ. vMotion не лише переміщує віртуальні машини, але й передає їхній стан та дані у реальному часі, забезпечуючи неперервну доступність.

VMware vSphere Fault Tolerance - це функціональність, яка забезпечує безперервну роботу віртуальних машин ВМ у випадку відмови апаратного забезпечення або програмних збоїв на рівні сервера. Fault Tolerance надає можливість створювати дублікат (дублююча ВМ) для основної віртуальної машини (основна ВМ) в реальному часі. Це дозволяє надійно забезпечити безперервність роботи у випадку відмови основної ВМ, оскільки дублююча ВМ безперервно відтворює той самий стан, що і основна ВМ [5].

Зміни, які відбуваються на основній ВМ, автоматично відображаються на дублюючій ВМ у реальному часі. Це досягається завдяки синхронній реплікації даних та стану системи між основною та дублюючою ВМ.

У випадку відмови основної ВМ, дублююча ВМ миттєво стає активною, забезпечуючи продовження роботи без будь-яких втрат даних або перерв у обслуговуванні. Функціональність Fault Tolerance налаштовується та керується через інтерфейс користувача VMware vSphere.

Хоча технологія Fault Tolerance забезпечує високий рівень надійності та доступності, важливо враховувати, що вона може призвести до збільшення використання ресурсів, особливо процесорного часу, через необхідність підтримувати дві копії ВМ одночасно. VMware Fault Tolerance дозволяє підтримувати безперервну роботу критичних додатків та сервісів, забезпечуючи автоматичне переключення на дублюючу ВМ у випадку відмови основної, що допомагає уникнути втрати даних та забезпечити неперервність бізнес-процесів.

Варто зазначити що VMware vSphere не є безкоштовним продуктом. Він має комерційну ліцензійну модель, яка передбачає плату за використання програмного забезпечення. Ліцензійні витрати варіюються в залежності від обраної версії, функцій та кількості фізичних процесорів на сервері. VMware пропонує різні рівні ліцензування та пакети продуктів, які мають свої власні функціональні можливості, обмеження та цінові плани. Це може включати такі варіанти як стандартні пакети, підписки на річний термін, пакети для малих підприємств тощо.

Однак VMware також має безкоштовні для використання продукти, такі як VMware ESXi, який є безкоштовним гіпервізором для використання на одному фізичному сервері, але має сильно обмежені можливості порівняно з платними версіями.

Отже, хоча є безкоштовні варіанти від VMware, повноцінний набір функцій та розширених можливостей VMware vSphere вимагає придбання комерційної ліцензії від компанії.

1.4. Огляд Hyper-V від Microsoft

Microsoft пропонує рішення віртуалізації під назвою Hyper-V, яке є ключовим компонентом їх платформи для віртуалізації і забезпечує віртуалізацію на рівні сервера для користувачів Windows [6].

Hyper-V від Microsoft - це гіпервізор, який можна класифікувати як типу 1, оскільки він працює безпосередньо на апаратному рівні. Це означає, що Hyper-V управляє апаратними ресурсами напряму та надає можливість створювати та управляти віртуальними машинами.

Гіпервізор має ряд функцій, включаючи міграцію віртуальних машин, підтримку клонування та шаблонів, можливість резервного копіювання та відновлення, а також можливості моніторингу та управління через Hyper-V Manager. Hyper-V інтегрований з іншими продуктами та сервісами Microsoft, такими як Active Directory, що робить його привабливим рішенням для середовищ на основі Windows. Hyper-V входить у склад безлічі продуктів Microsoft, включаючи Windows Server, тому для багатьох користувачів він може бути доступний без додаткових витрат. Він підтримує роботу на системах з багатьма процесорами, що робить його гнучким у різних розмірах інфраструктури.

Microsoft Hyper-V Failover Cluster - це технологія, яка дозволяє створювати високодоступні віртуальні середовища на базі Hyper-V [7]. Ця технологія надає можливість автоматичного відновлення роботи ВМ у разі відмови апаратного

забезпечення чи програмного збою, забезпечуючи мінімальний час відмови та безперервну доступність сервісів.

Кластер Hyper-V забезпечує високий рівень доступності для віртуальних машин. Якщо один з вузлів кластера або апаратне обладнання відмовляє, віртуальні машини автоматично переносяться на інші вузли кластера без втрати продуктивності. Кластер Hyper-V Failover Cluster може включати кілька вузлів, що дозволяє розширювати обчислювальні ресурси та забезпечувати більшу масштабованість. Управління кластером та його ресурсами може здійснюватися через інструменти управління, такі як Failover Cluster Manager, які дозволяють моніторити стан кластера та виконувати адміністративні завдання.

Цей клас рішень використовується для створення віртуальних інфраструктур що забезпечують високу доступність та надійність віртуальних середовищ. Microsoft Hyper-V Failover Cluster дозволяє створювати високодоступні віртуальні інфраструктури, які можуть забезпечити неперервну роботу під час відмови окремих елементів апаратного забезпечення чи програмного збою.

Hyper-V Replica - це функція у Hyper-V кластері, яка забезпечує можливість створення резервних копій ВМ та реплікацію їх даних на інший Hyper-V сервер (зазвичай в іншому розташуванні) для забезпечення високої доступності та відновлення в разі відмови. Це дозволяє відновлювати роботу віртуальних машин у випадку відмови апаратного забезпечення або інших подій.

Реплікація відбувається асинхронно, тобто зміни даних на первинному сервері передаються на вторинний сервер з певним затримкою. Це дозволяє знизити вплив на продуктивність віртуальних машин під час реплікації. Адміністратор може налаштувати частоту змін, які реплікуються на вторинний сервер, щоб забезпечити бажану точність відновлення даних. Коли відновлення є необхідним, Hyper-V Replica дозволяє адміністраторам вибирати точку відновлення та виконувати відновлення віртуальних машин.

Hyper-V Replica дозволяє підвищити надійність віртуальної інфраструктури шляхом регулярного створення резервних копій та реплікації даних на

вторинний сервер, що дозволяє швидко відновити роботу системи в разі виникнення проблем.

Live Migration в Hyper-V - це функціональність, що дозволяє переміщувати робочий стан віртуальної машини між різними фізичними серверами без вимкнення віртуальної машини або призупинення роботи в ній. Це дозволяє адміністраторам управляти ресурсами сервера, оптимізувати роботу та забезпечувати високу доступність системи. Під час процедури ресурси, такі як пам'ять, обчислювальна потужність та стан процесора, активно мігрують з одного сервера на інший, забезпечуючи безперервну продуктивність. Адміністратори можуть використовувати живу міграцію для рівномірного розподілу ресурсів між серверами, щоб зменшити навантаження та оптимізувати використання обчислювальних потужностей. Live Migration у Hyper-V допомагає забезпечити високий рівень доступності та надійності віртуальної інфраструктури, дозволяючи переміщати робочий стан віртуальних машин без відмови у роботі.

Також варто зазначити що цінова та ліцензійна політика для Microsoft Hyper-V визначається кількома факторами, такими як версія Windows Server, тип ліцензування та пакети функцій. Hyper-V є частиною Windows Server, який доступний у різних редакціях, таких як Standard, Datacenter тощо. Ліцензійні витрати можуть залежати від обраної версії Windows Server.

Microsoft має безкоштовну версію Hyper-V, яка називається Microsoft Hyper-V Server. Hyper-V Server - це гіпервізор, який надає базовий функціонал для віртуалізації та управління віртуальними машинами.

Однак, в порівнянні з повноцінною версією Windows Server, Hyper-V Server має обмежені можливості, такі як менше інтегрованих інструментів управління та підтримка розширених функцій. Hyper-V Server може бути привабливим вибором для організацій або користувачів, які шукають безкоштовне вирішення віртуалізації і не потребують всіх функцій, які надаються в повноцінних версіях Windows Server. Хоча у Hyper-V можуть бути обмеження порівняно з іншими рішеннями віртуалізації, він залишається важливою опцією для тих, хто працює

з середовищами Microsoft, оскільки він інтегрований у їх екосистему та може бути доступний для використання як частина платформи Windows Server.

1.5. Огляд технології віртуалізації XEN від Citrix

Xen - це гіпервізор з відкритим кодом, розроблений в університеті Кембриджа та зараз розвивається спільнотою розробників із усього світу. Citrix є однією з ключових компаній, які сприяють розвитку цієї технології, пропонуючи продукти, побудовані на основі Xen [8].

Xen є гіпервізором типу 1, що означає, що він працює безпосередньо на апаратному рівні. Це сприяє ефективному використанню апаратних ресурсів та підвищує продуктивність. Xen може віртуалізувати різні операційні системи, включаючи Linux та Windows, дозволяючи запускати різні ОС на одному фізичному сервері. Гіпервізор використовує техніки паравіртуалізації, які дозволяють гостьовим операційним системам знати про те що вони працюють в віртуалізованому середовищі та оптимізувати продуктивність. Через свою архітектуру та підтримку паравіртуалізації Xen забезпечує високу продуктивність віртуальних машин.

Citrix Hypervisor є продуктом компанії Citrix, побудованим на основі технології гіпервізора Xen. Це комерційне віртуалізаційне рішення, яке надає широкі можливості віртуалізації для об'єднання фізичних ресурсів сервера та управління віртуальними машинами. Citrix надає інструменти управління віртуальними машинами, включаючи можливість моніторингу, масштабування та автоматизації завдань. Підтримує функції безпеки для віртуальних середовищ, такі як ізоляція ресурсів та захист від зовнішніх загроз.

Citrix Hypervisor надає можливості міграції та високої доступності для забезпечення надійності та безперервності роботи віртуальних машин у віртуальному середовищі. Функція Live Migration дозволяє адміністраторам переміщати робочий стан віртуальної машини з одного фізичного сервера на інший без відмови в роботі. Це дозволяє балансувати навантаження, робити

планове обслуговування серверів або реагувати на потреби в ресурсах без перерви для користувачів або послуг. Функція HA забезпечує автоматичне відновлення роботи віртуальних машин у випадку відмови апаратного забезпечення. Якщо фізичний сервер перестає працювати, віртуальні машини, що на ньому працювали, автоматично перезапускаються на іншому доступному сервері, забезпечуючи таким чином найбільшу доступність послуги.

Ці функції спрямовані на забезпечення надійності та безперервності в роботі віртуальних середовищ, що є критично важливим для бізнес-середовищ, де необхідна висока доступність та надійність інфраструктури.

XenCenter є графічним інтерфейсом користувача для управління і контролю гіпервізором Xen. Це інструмент, який надає широкий спектр функцій для адміністрування віртуальних машин, інфраструктури, а також ресурсів інфраструктури віртуалізації.

Xen Orchestra - це веб-інтерфейс для управління і моніторингу гіпервізором Xen. Це рішення, розроблене спільнотою, яке надає розширений набір функцій для керування віртуальними машинами та інфраструктурою віртуалізації.

Citrix пропонує кілька рівнів ліцензування, кожен з яких має власні функції та обмеження. Базова версія Citrix Hypervisor є безкоштовну та має обмежені функції, але надає базовий функціонал для віртуалізації. Ця версія може задовольнити потреби невеликих підприємств або використовуватися для тестування/розробки. Платна версія має розширені можливості, такі як технологія HA та інші функції. Вона призначена для підприємств середнього розміру, які потребують розширеного функціоналу.

1.6. Огляд реалізації KVM від Oracle

KVM - це гіпервізор типу 1, який вбудований у ядро Linux і працює безпосередньо з апаратним забезпеченням сервера, використовуючи функції вбудованої апаратної віртуалізації процесорів Intel VT або AMD-V [9].

Гіпервізор підтримує віртуалізацію різних операційних систем, таких як Linux, Windows, Unix та інші.

Oracle Linux Virtualization Manager є рішенням для управління віртуалізацією, яке базується на технології KVM та розроблене компанією Oracle [10]. Це інструмент управління віртуальними машинами, який надає можливості для створення та управління віртуальними середовищами на базі Oracle Linux. Це рішення дозволяє адміністраторам створювати, розгортати, керувати та моніторити віртуальні машини з графічним інтерфейсом. Надає можливість переміщення віртуальних машин між фізичними серверами без перерви у роботі. Забезпечує можливість створення шаблонів віртуальних машин для швидкого розгортання. Надає інструменти для моніторингу використання ресурсів та продуктивності віртуальних машин. Дане рішення повністю інтегрується з операціною системою Oracle Linux.

OLVM має можливості для забезпечення високої доступності в віртуальному середовищі. Ця функція дозволяє зменшити можливі відмови та забезпечити безперервну роботу віртуальних машин у разі виникнення проблем. Віртуальні машини можуть бути організовані в кластери, де забезпечується спільний доступ до ресурсів та вирішення проблем у випадку відмови одного з серверів. У разі відмови фізичного сервера, на якому працює віртуальна машина, механізм HA може автоматично перезапустити цю машину на іншому фізичному сервер в кластері з мінімальною перервою у роботі [11].

Система постійно відстежує стан серверів та віртуальних машин, виявляє відмови та вживає відповідних заходів, щоб уникнути втрати даних або недоступності сервісів. Забезпечує можливість створення резервних копій віртуальних машин та відновлення їх у разі потреби, що дозволяє відновити роботу системи після відмови. OLVM пропонує інструменти для реалізації високої доступності в віртуальному середовищі, щоб забезпечити неперервну роботу бізнес-систем та уникнути відмов в разі проблем з апаратурою чи програмним забезпеченням.

OLVM - це безкоштовне рішення для віртуалізації, яке доступне для завантаження та використання. Воно має відкрите програмне забезпечення та безкоштовний код. Однак для офіційної технічної підтримки потрібна комерційна підписка від Oracle.

1.7. Порівняння кластерів віртуалізації

Порівняння кластерів віртуалізації різних постачальників, таких як VMware, Microsoft, Citrix, та Oracle, є корисним для розуміння їхніх особливостей та можливостей. Платформи віртуалізації володіють власними особливостями та перевагами. Вибір певної віртуалізаційної платформи залежить від конкретних потреб організації, бюджету, інфраструктури, вимог до функціональності та підтримки. У таб.1.1 наведено порівняння основних можливостей та характеристик кластерів віртуалізації різних постачальників.

Таблиця 1.1

Порівняння характеристик кластерів віртуалізації

Характеристика	VMware ESXi Cluster	Microsoft Hyper-V Cluster	Citrix Hypervisor Cluster	Oracle Linux KVM Cluster
Тип гіпервізора	Тип 1	Тип 1	Тип 1	Тип 1
Підтримувані операційні системи ВМ	Windows, Linux, Unix, MacOS	Windows, Linux, Unix	Windows, Linux, Unix	Windows, Linux, Unix
Інтеграція з іншими продуктами	Так	Так	Так	Так
Функції Live Migration ВМ	Так	Так	Так	Так
Система управління та моніторингу	Так	Так	Так	Так
Масштабованість	Так	Так	Так	Так
Функції високої доступності	High Availability, Fault Tolerance	High Availability	High Availability	High Availability
Ліцензування	Так	Так	Так	Ні

VMware кластер надає високу ступінь автоматизації та масштабованості, різноманітні функції управління, включаючи vMotion, HA, DRS та інші. Широко використовується в індустрії, має велику базу користувачів та розробників, підтримується багатьма сторонніми рішеннями. Що до вартості, то порівняно з решту це найдорожче рішення.

Кластер на базі Hyper-V також підтримує схожі функції, такі як Live Migration, Failover Clustering, Hyper-V Replica, та інші. Також він інтегрований з іншими продуктами Microsoft, такими як Active Directory, що забезпечує спрощене управління. Але також не є безкоштовний продуктом та вимагає ліцензування.

Віртуалізаційний кластер на основ Citrix Hypervisor має подібні можливості до решти, включаючи Live Migration, HA, Disaster Recovery та інші. Також він є платним.

Кластери на основі Oracle Linux KVM забезпечує можливості високої доступності, Live Migration, масштабування та інші функції віртуалізації. Інтегрується з іншими рішеннями Oracle. Oracle Linux використовує відкритий код, що дозволяє спільноті розробників та користувачів активно співпрацювати, вносити зміни та вдосконалювати функціонал платформи. Для управління кластером KVM використовується OLVM, який надає зручний інтерфейс для управління віртуальними машинами, моніторингу та керування ресурсами.

Oracle Linux є дистрибутивом Linux, який доступний для завантаження та використання безкоштовно. Також OLVM є безкоштовним інструментом управління віртуалізацією на основі KVM. Це є привабливим рішенням для малих та середніх підприємств з обмеженими бюджетами, які шукають стабільну та безкоштовну операційну систему та високоякісне програмне забезпечення без додаткових витрат на ліцензії.

1.8. Висновки до розділу

У розділі проведено огляд та порівняння сучасних методів створення кластерів на основі платформ віртуалізації VMware ESXi, Microsoft Hyper-V, Citrix Hypervisor та Oracle Linux KVM. Досліджено їхні можливості у забезпеченні високої доступності системи у випадку відмови одного чи кількох серверів. Під час аналізу детально розглянута функціональність кожної технології віртуалізації та їхні можливості забезпечити надійність та безвідмовність інфраструктури. Усі чотири виробники платформ віртуалізації мають схожі можливості створення високодоступних систем, але лише рішення на базі KVM не потребує додаткових фінансових витрат на ліцензії. Тому було обрано Oracle Linux як засіб для створення кластеру високої доступності.

РОЗДІЛ 2
ПРИНЦИПИ ТА ОСОБЛИВОСТІ РОБОТИ ГІПЕРВІЗОРА KVM

2.1. Принцип роботи гіпервізора KVM

Гіпервізор KVM - це віртуальний стек для ядра Linux, що дозволяє створювати та управляти віртуальними машинами на платформі Linux.

На рис.2.1 показано архітектуру віртуалізації на основі KVM.

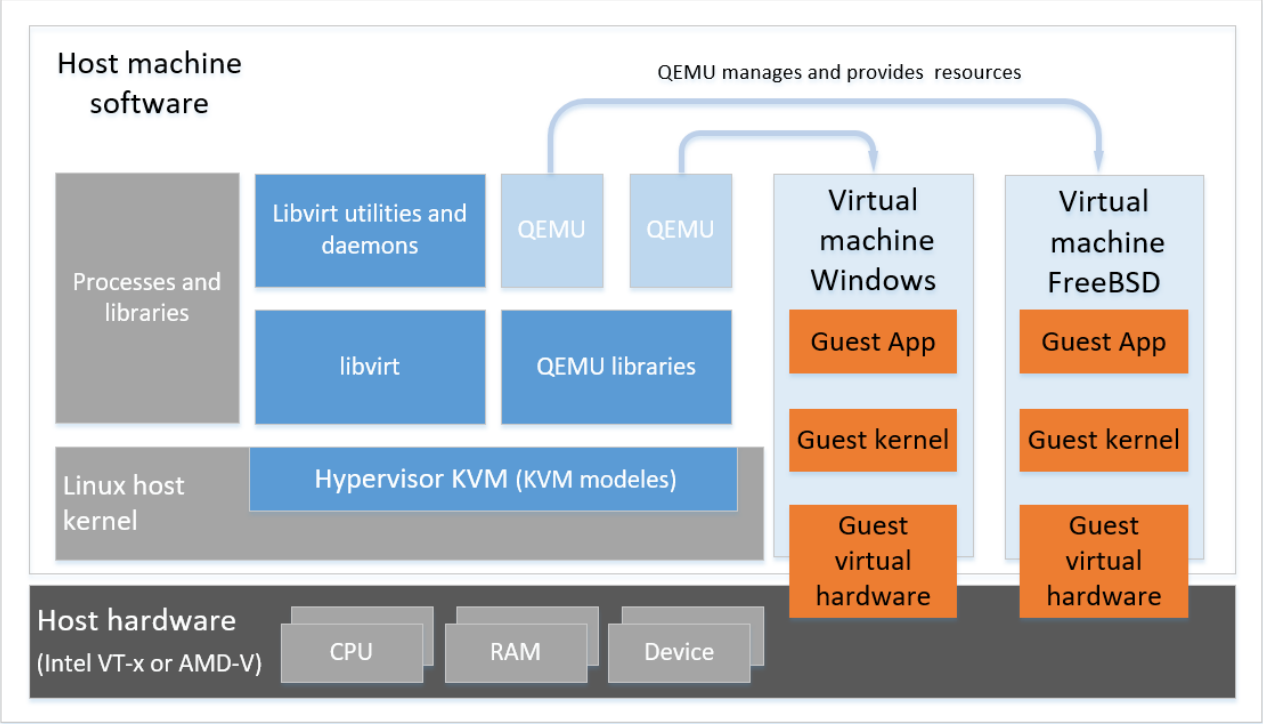


Рис. 2.1. Архітектура віртуалізації на базі KVM

Основна ідея полягає в тому, що гіпервізор KVM, який керує віртуальними машинами, запускається безпосередньо на апаратному забезпеченні фізичного сервера і взаємодіє з операційною системою як частина її ядра.

Цей підхід надає відмінну продуктивність та ефективність, оскільки гіпервізор має безпосередній доступ до ресурсів сервера. Операційна система, на якій працює гіпервізор, має роль виконавчої системи для віртуальних машин та управляє взаємодією з апаратним забезпеченням.

Модуль KVM та драйвери ядра віртуалізації забезпечують основні ресурси для віртуалізації програмного забезпечення простору користувача на хост-системі з ядром Linux. На рівні простору користувача, емулятор QEMU моделює повну віртуалізовану апаратну платформу, на якій може працювати гостьова операційна система, і управляє розподілом ресурсів на хості, що потрібні для гостьової системи.

Крім цього, QEMU взаємодіє з KVM, передаючи інструкції, і KVM гарантує належне призначення ресурсів ядра, необхідних для виконання цих інструкцій. Результатом цього є можливість QEMU вносити зміни в простір користувача, такі як створення чи зміна віртуальної машини або виконання дій у гостьовій операційній системі віртуальної машини

Додатково, пакет програмного забезпечення libvirt виступає як рівень управління та комунікації, спрощуючи взаємодію з QEMU. libvirt забезпечує дотримання правил безпеки та надає різноманітні інструменти для налаштування та запуску віртуальних машин.

Гіпервізор KVM використовує апаратну віртуалізацію (Intel VT-x або AMD-V), яка дозволяє оптимально використовувати можливості процесора для віртуалізації. Управління ресурсами у гіпервізорі KVM включає в себе ряд інструментів та можливостей для ефективного розподілу та контролю ресурсів сервера між віртуальними машинами.

Гіпервізор дозволяє призначати віртуальним машинам окремі ядра центрального процесора, забезпечуючи кожній віртуальній машині відведену обчислювальну потужність. Керування пам'яттю дозволяє виділяти та контролювати обсяг оперативної пам'яті для кожної віртуальної машини. Можливе динамічне збільшення або зменшення обсягу виділеної пам'яті. Контроль доступу до пристроїв вводу-виводу (I/O), таких як диски, мережеві інтерфейси, дозволяє ефективно виділяти та керувати доступом до цих ресурсів для віртуальних машин. Гіпервізор KVM надає засоби для створення, розширення та управління віртуальними дисками та файловими системами для віртуальних машин. Також дозволяє ефективно керувати та виділяти ресурсами

фізичного сервера для віртуальними машинами, надаючи їм доступ до обчислювальної потужності, пам'яті та інших ресурсів. KVM підтримує віртуалізацію мережі, що дозволяє створювати віртуальні мережі та управляти ними для забезпечення комунікації між віртуальними машинами, а також зовнішніми мережами. Гіпервізор надає можливість створення віртуальних комутаторів (virtual switches), які дозволяють віртуальним машинам спілкуватися між собою та зовнішніми мережами. Ці віртуальні комутатори можуть мати свої правила маршрутизації, фільтрації пакетів, та інші налаштування, аналогічні функціям фізичних мережевих комутаторів. Також KVM підтримує віртуальні локальні мережі (VLAN), що дозволяє групувати віртуальні машини в різні сегменти мережі, незалежно від їх фізичної розташування. Це спрощує управління мережею та забезпечує безпеку та гнучкість. KVM може використовувати мережеві мости (bridge networking) для з'єднання віртуальних машин з фізичною мережею. Це дозволяє віртуальним машинам отримувати доступ до мережевих ресурсів та взаємодіяти з зовнішніми пристроями і також дозволяє отримати доступ по мережі до віртуальних машин.

Гіпервізор KVM підтримує різні операційні системи (ОС) як гостьові системи. KVM добре підтримує Linux-сумісні операційні системи. ОС з ядром Linux можна легко використовувати в якості гостьових ОС на KVM. Також є підтримка для гостьових операційних систем Windows, включаючи різні версії Windows Server та Windows Desktop, такі як Windows 10 або Windows 11. Крім Linux і Windows, KVM може підтримувати інші ОС, такі як FreeBSD, Solaris, а також інші варіанти UNIX або спеціалізовані ОС, які можуть бути встановлені як гостьові системи. Гіпервізор KVM зазвичай не обмежений певними версіями чи дистрибутивами ОС, які можна встановити як гостьові. Він може підтримувати широкий спектр версій різних операційних систем. Завдяки цій розширеній підтримці різних операційних систем, KVM є популярним вибором для створення різноманітних віртуальних інфраструктур, де потрібна підтримка різних ОС для різних завдань або додаткових сервісів.

Гіпервізор KVM дозволяє працювати з різними сховищами (локальне сховище, NFS, iSCSI та GlusterFS) для забезпечення зберігання та обміну даними між віртуальними машинами.

Локальне сховище використовується для зберігання віртуальних дисків на фізичних пристроях сервера. Це може бути звичайний диск (HDD або SSD), на якому зберігається віртуальна машина. KVM може працювати зі сховищами, які надають доступ до файлової системи через NFS, яка дозволяє розподілений доступу до файлів, тому віртуальні машини можуть звертатися до файлових ресурсів через мережу. Протокол iSCSI дозволяє віртуальним машинам використовувати зовнішні блочні пристрої як віртуальні диски. Це дозволяє використовувати зовнішні сховища, які підключені до мережі через iSCSI. Розподілена файлова система GlusterFS дозволяє об'єднувати різні простори зберігання в мережу, створюючи єдиний великий пул зберігання даних. KVM може використовувати GlusterFS для зберігання віртуальних дисків.

Ці різні методи зберігання даних надають можливість вибору при налаштуванні сховищ для віртуальних машин у залежності від потреб інфраструктури.

2.2. Апаратна віртуалізація Intel VT-x та AMD-V

У сучасних процесорах Intel та AMD з'явилися розширення віртуалізації, такі як Intel VT-x та AMD-V, які стали необхідними для ефективного використання ресурсів та розвитку віртуалізації [12]. Ці технології дозволяють створювати віртуальні середовища на одному фізичному сервері, забезпечуючи кращу ізоляцію, управління та використання ресурсів.

Існують різні типи віртуалізації, включаючи повну, паравіртуалізацію та апаратну віртуалізацію. Технології, такі як Intel VT-x і AMD-V, відносяться до категорії апаратної віртуалізації, що сприяє підвищенню ефективності та безпеки віртуалізованих оточень.

Технологія віртуалізації Intel VT-x представляє собою набір вдосконалень апаратного забезпечення, спрямованих на спрощення створення та управління віртуальними машинами. Вона включає в себе різноманітні функції, такі як VMX і EPT, які покликані оптимізувати продуктивність віртуалізованих обчислювальних завдань.

EPT - це технологія віртуалізації, яка використовується процесорами Intel для підтримки віртуальних машин та оптимізації роботи гіпервізорів. Ця технологія також відома як Second Level Address Translation (SLAT). EPT сприяє покращенню продуктивності віртуальної машини, дозволяючи їм працювати більш ефективно та швидше. Основна мета EPT - це оптимізувати процес перетворення віртуальних адрес в реальні фізичні адреси шляхом використання додаткових таблиць перекладу, які знаходяться на рівні апаратних ресурсів процесора. EPT дозволяє прискорити перетворення віртуальних адрес в реальні фізичні адреси, що пришвидшує доступ до пам'яті та покращує продуктивність віртуальних машин. Завдяки оптимізації процесу перетворення адрес, віртуальні машини виконують операції швидше та з меншими витратами ресурсів процесора. EPT дозволяє збільшити кількість віртуальних машин, які можуть працювати на одній фізичній системі, без втрати продуктивності.

VMX є технологічним розширенням архітектури процесорів Intel. Це розширення відноситься до набору інструкцій, які використовуються для підтримки віртуалізації на рівні апаратури. Розширення надає підтримку для технологій віртуалізації, дозволяючи розподіляти апаратні ресурси для оптимальної роботи гіпервізорів. VMX забезпечує ряд інструкцій процесора, які дозволяють гіпервізорам безпосередньо керувати віртуалізованими середовищами, забезпечуючи обробку привілеєвих інструкцій та використання апаратних ресурсів для віртуалізованих операційних систем.

VT-x може працювати в двох режимах: кореневому режимі VMX (root mode) і некореневому режимі VMX (non-root mode).

На Рис.2.2 показана модель ЦП і пам’яті з віртуалізацією процесів та VMX, яка повністю віртуалізовує ЦП в апаратному забезпеченні.

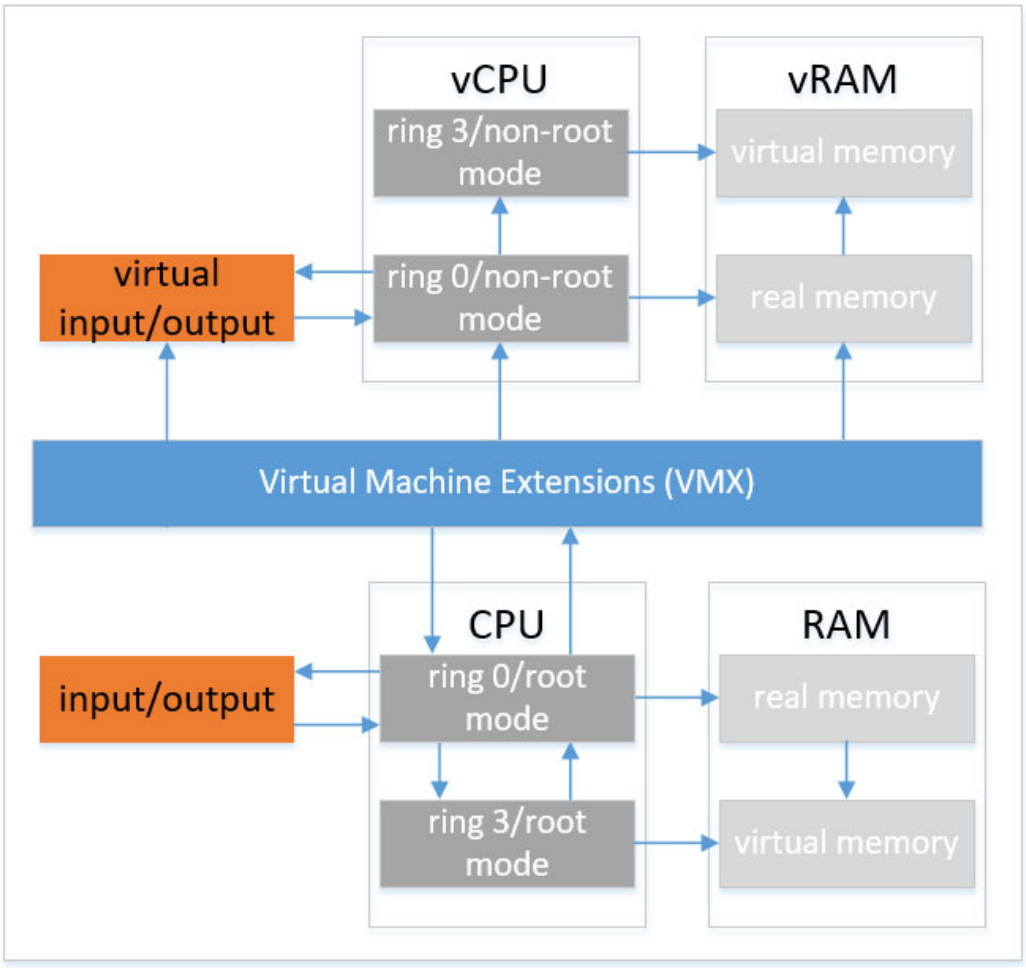


Рис. 2.2. Модель ЦП та пам’яті з використанням VMX

У режимі root mode процесор працює як традиційні процесори попередніх поколінь, які не мають підтримки технології віртуалізації VT-x. В цьому режимі існує чотири рівні привілеїв, які називаються кільцями (rings), і підтримується аналогічний набір інструкцій з додаванням деяких, що призначені для реалізації віртуалізації. Root mode використовується операційною системою хоста без використання віртуалізації, і також використовується гіпервізором, коли активовано віртуалізацію.

У режимі non-root відбуваються суттєві зміни у роботі центрального процесора. Додатково існують ще чотири рівнів привілеїв, а також застосовується той самий набір інструкцій (див. рис.2.3).

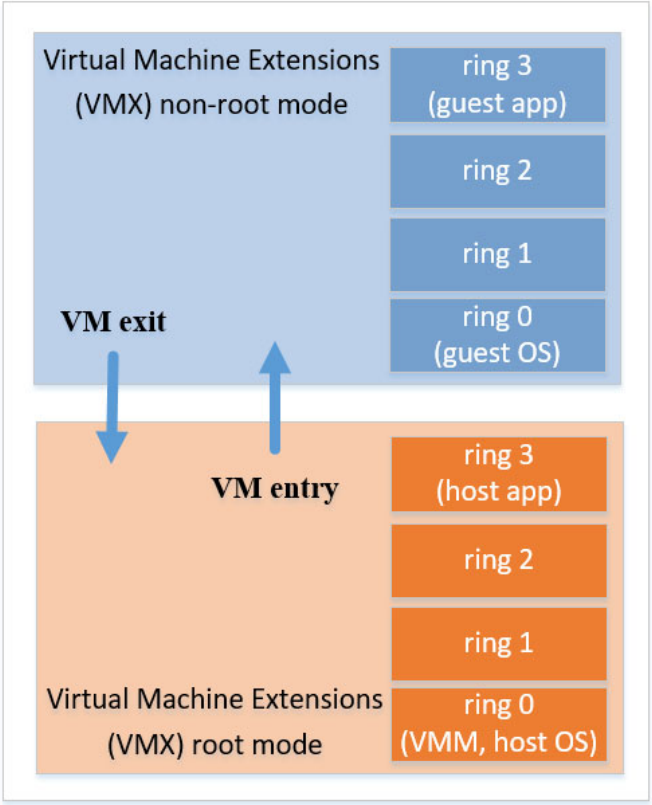


Рис. 2.3. Root та non-root режими роботи VMX

В режимі non-root існує нова структура, відома як VMCS. Ця структура відповідає за контроль роботи центрального процесора та визначає поведінку певних інструкцій. У цьому режимі працюють гостьові системи.

Перехід з режиму root до non-root відомий як VM entry, а повернення назад - як VM exit. VMCS містить гостьову та хост-зону стану, яка зберігається та відновлюється під час входу та виходу з віртуальної машини. Одне з ключових завдань VMCS - це контроль того, які гостьові операції призведуть до виходу з віртуальної машини. Кожна гостьова операційна система має свій власний VMCS.

VMCS надає гіпервізору точний контроль над тим, які операції можуть виконуватись гостевою операційною системою. Наприклад, гіпервізор може дозволити гостьовій системі записувати певні біти в затінені керуючі регістри, але лише у дозволених областях. Це забезпечує ефективну віртуалізацію, коли гостям можна дозволити записувати керуючі біти, не перешкоджаючи роботі гіпервізора, тим самим уникнувши змін, які можуть вплинути на функціонування

гіпервізора. VMCS також регулює доставку переривань і винятків, що дозволяє гіпервізору контролювати ці процеси для кращого управління віртуалізованою системою. Коли виникає подія або інструкція, яка спричиняє VM exit, VMCS зберігає інформацію про причину виходу, що дозволяє гіпервізору зрозуміти, чому було призначено вихід із віртуальної машини. Це надає гіпервізору засоби ефективно взаємодіяти зі станом і подіями, що відбуваються у віртуальному середовищі.

Щодо управління режимом VT-x, гіпервізор може активувати або деактивувати цей режим, використовуючи інструкції VMXON і VMXOFF відповідно. Це дозволяє гіпервізору управляти включенням та виключенням апаратної підтримки віртуалізації на рівні процесора, забезпечуючи контроль над віртуалізованим середовищем.

На рис.2.4 показано роботу гапервізора з використанням VMCS.

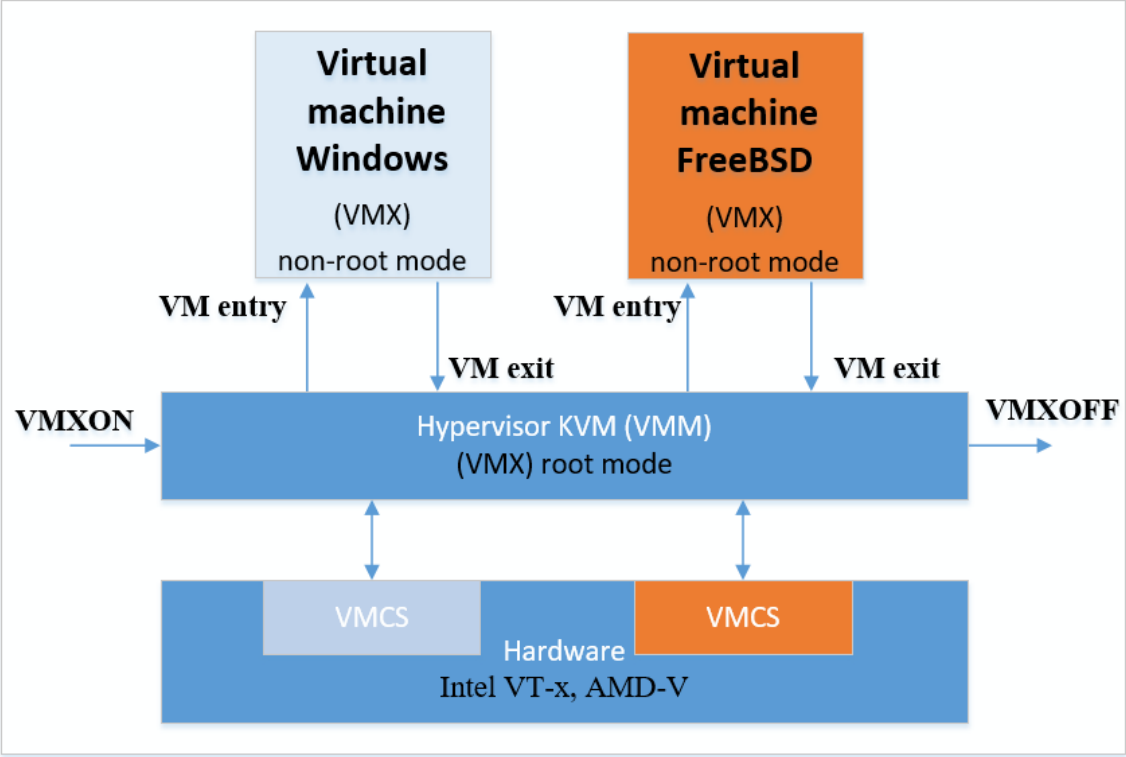


Рис. 2.4. Гіпервізор KVM з використанням VMCS

При використання апаратної віртуалізації за допомогою Intel VT-x або AMD-V, гіпервізор запускається за допомогою інструкції VMXON, що дозволяє перейти у режим роботи VMX. Далі, гіпервізор використовує інструкції VMLAUNCH або VMRESUME для створення операції VM Entry і входу в гостьову операційну систему. У цьому випадку, процесор знаходиться в режимі non-root.

Коли гостьова операційна система виконує привілейовані операції, які призводять до виходу з віртуальної машини VM exit, процесор перемикається в режим root. Гіпервізор обробляє подію VM exit, а потім виконує інструкцію VM Entry для продовження роботи гостьової операційної системи. Гіпервізор може вийти з режиму роботи VMX, використовуючи інструкцію VMXOFF, щоб завершити роботу з віртуалізацією.

Гостьова операційна система, яка працює в режимі non-root у віртуальному середовищі, має свій власний адресний простір, який повністю ізольований від інших віртуальних машин та від гіпервізора. Це відокремлення адресних просторів запобігає конфліктам та спільному використанню ресурсів.

Крім того, код ядра гостьової операційної системи виконується на найбільш привілейованому рівні (кільці 0) у режимі non-root віртуальної машини з підтримкою технології віртуалізації (див.рис.2.3). Це дозволяє виконання ядра гостьової ОС без втручання у роботу менш привілейованих рівнів.

Важливою особливістю є те, що доступ до операцій введення-виведення (I/O) з гостьового коду також викликає VM exit, навіть якщо він виконується на кільці 0 у режимі non-root VMX. Це дозволяє гіпервізору емулювати роботу різних пристроїв та забезпечувати ізольованість від реальних пристроїв, що забезпечує безпеку та контроль над взаємодією гостьових операційних систем з апаратним забезпеченням.

Отже, в root режимі VMX гіпервізор, працює як контролер, керуючи та моніторячи роботу віртуальних машин та взаємодіючи з апаратним забезпеченням. Головна функція гіпервізора - забезпечення ізоляції та ресурсів між різними віртуальними середовищами.

У non-root режимі VMX виконується гостьова операційна система в межах віртуальної машини. Гіпервізор у цьому випадку стежить за віртуалізованим обсягом ресурсів, такими як процесорний час, пам'ять та введення-виведення, та керує доступом до апаратних ресурсів реального сервера для гостьової операційної системи. Це розділення функцій дозволяє керувати віртуальними машинами ефективно та безпечно.

Подібно до Intel VT-x, AMD-V є апаратною технологією віртуалізації, яка надає спеціальні можливості для ефективної роботи віртуальних середовищ на процесорах AMD.

Однією з основних можливостей AMD-V є SVM, яка відповідає за підтримку безпеки віртуальних машин. Крім того, AMD-V також підтримує NPT, яка є подібною до EPT у Intel VT-x.

Основна мета SVM полягає в забезпеченні підвищеного рівня безпеки віртуальних машин. Ця функція спрямована на запобігання неправильному або несанкціонованому доступу віртуальних машин до інших областей системи або до операційної системи хоста.

NPT функція, схожа на EPT у процесорах Intel, яка дозволяє оптимізувати переклад віртуальних адрес у фізичні адреси та підвищує продуктивність шляхом оптимізації процесу адресації пам'яті.

Віртуалізація з підтримкою апаратного забезпечення відіграє ключову роль у забезпеченні ефективної та оптимізованої роботи віртуальних середовищ. EPT від Intel та NPT від AMD це апаратні функції, які дозволяють використовувати вкладені таблиці сторінок. Вони оптимізують трансляцію віртуальної адреси у фізичну, що значно покращує продуктивність віртуальної машини. Ці функції спрощують доступ до пам'яті для віртуальних середовищ та зменшують накладні витрати, що допомагає виконувати операції з пам’яттю швидше та ефективніше. Завдяки цим функціям, віртуальна машина може більш точно та ефективно управляти доступом до фізичної пам'яті, що сприяє покращенню її продуктивності та забезпечує майже рідну продуктивність в порівнянні з не віртуалізованими системами.

Технології віртуалізації Intel VT-x та AMD-V грають важливу роль у забезпеченні безпеки віртуальних середовищ. Вони дозволяють віртуальним машинам працювати на одній фізичній системі, але в ізольованому від інших способі. Ця ізоляція є ключовим елементом з точки зору безпеки, оскільки вона запобігає втручанню та взаємодії між віртуальними машинами одна з одною та з хост-системою. Якщо одна віртуальна машина стає жертвою атаки або вірусу, ізольований характер технології дозволяє запобігти розповсюдженню проблеми на інші віртуальні машини або хост-систему.

2.3. Взаємодія KVM та QEMU

KVM та QEMU співпрацюють для надання покращеної функціональності віртуалізації на рівні ядра Linux (див. рис.2.1).

Як вже було сказано віртуальна машина на основі ядра - це розширення ядра Linux, яке перетворює основне ядро операційної системи в гіпервізор. Цей гіпервізор забезпечує роботу віртуальних машин на базі Linux.

У KVM можна виділити два основних типи модулів. Модуль kvm.ko - є базовим модулем, що завжди потрібен. Він надає базову функціональність для роботи з гіпервізором KVM. Цей модуль не залежить від конкретного типу архітектури процесора. Для роботи KVM потрібні модулі, які відповідають конкретним архітектурам процесорів. Наприклад, kvm-intel.ko використовується для процесорів Intel, а kvm-amd.ko - для процесорів AMD. Ці модулі надають підтримку конкретних можливостей віртуалізації, які притаманні кожній з архітектур. Установка правильних модулів залежить від архітектури процесора, який використовується на головній машині. Модуль забезпечить оптимальну підтримку функцій віртуалізації для цього конкретного типу процесора.

Взаємодія між KVM і QEMU відбувається через кілька механізмів, що забезпечують обмін даними та управління віртуальними машинами:

Файл пристрою /dev/kvm - це основний інтерфейс, який надає KVM для взаємодії з QEMU. Він підтримує набір IOCTL (Input/Output Control) системних

викликів, які дозволяють QEMU керувати віртуальними машинами та проводити з ними різноманітні операції, такі як створення, запуск, призупинення, відновлення та припинення віртуальних машин.

Відображені сторінки пам'яті (memory-mapped pages) - це механізм, що дозволяє відображення областей пам'яті, доступ до яких здійснюється через операції читання та запису як звичайний запис у файл. У віртуальному адресному просторі кожного процесу може бути пам'ять, яка відображається з або до файлів на диску чи в оперативну пам'ять.

При віртуалізації memory-mapped pages використовуються для ефективного обміну даними між гіпервізором (KVM) та віртуальною машиною (QEMU). Кожен віртуальний процесор (vCPU) віртуальної машини має дві такі сторінки пам'яті.

Вони дозволяють забезпечити швидкий та безпечний доступ до віртуальної пам'яті, що використовується в межах віртуальної машини, забезпечуючи можливість обміну даними між QEMU та віртуальною машиною, яка працює в ядрі KVM. Цей механізм використовується для передачі великих обсягів даних та забезпечення ефективності роботи віртуальної системи.

У цих способах взаємодії QEMU функціонує як емулятор, який надає інтерфейс користувача та обробляє взаємодію з гіпервізором KVM. KVM, зі свого боку, допомагає QEMU управляти та взаємодіяти з віртуальними машинами на рівні ядра операційної системи [13].

Інтерфейс API /dev/kvm має трирівневу ієрархію.

Глобальний рівень управління використовується для зміни глобальних параметрів KVM, таких як управління пам'яттю, створення та видалення віртуальних машин.

Рівень віртуальних машин забезпечує можливість взаємодії з конкретною віртуальною машиною. Ці виклики включають управління віртуальними ЦП, пам'яттю та іншими ресурсами цієї машини.

API найнижчого рівня стосується управління окремими віртуальними процесорами (vCPU) в межах віртуальної машини. Він дозволяє створювати та

керувати цими віртуальними процесорами через відповідні виклики API. Дане API є найближчим до рівня обслуговування пристрою /dev/kvm. Кожен vCPU віртуальної машини, створеної через KVM, має свій окремий контекст та може керуватися за допомогою відповідних викликів API.

Коли QEMU створює віртуальний процесор, він працює в тому самому потоці, що й віртуальний процесор, і взаємодіє з ним через ioctl та memory-mapped pages (відображені сторінки пам'яті), які надають доступ до низькорівневих операцій з керування віртуальними ресурсами. Відображення пам'яті між QEMU та vCPU використовується для передачі великих обсягів даних та спільної роботи між цими складовими віртуальної машини. QEMU також працює, як програма простору користувача та забезпечує емуляцію та віртуалізацію. В режимі емуляції QEMU імітує апаратне забезпечення цільової системи, що дозволяє запускати програми і операційні системи, призначені для цільової архітектури (наприклад ARM), на різних хост-системах (наприклад Intel) без вимог до підтримки цільової архітектури на хост системі.

Отже, QEMU може підробляти обладнання шляхом емуляції конкретних архітектур процесора чи пристроїв. Коли викристовується QEMU для створення віртуальної машини з певною архітектурою (наприклад ARM), QEMU емулює цю архітектуру для програмного забезпечення, яке запускається всередині цієї віртуальної машини.

TCG є ключовою складовою QEMU для ефективної емуляції. TCG перетворює байт-код, який отримується в результаті інтерпретації інструкцій, на оптимізований машинний код, який потім виконується на апаратурі хоста. Це підвищує продуктивність емуляції, оскільки машинний код може виконуватися швидше, ніж інтерпретований байт-код. TCG допомагає QEMU ефективно виконувати інструкції, які надходять від віртуальних машин, на апаратурі хоста. Це забезпечує ефективність емуляції та забезпечує віртуальним машинам можливість працювати на хостовій системі, незважаючи на їхню архітектуру.

У режимі віртуалізації QEMU співпрацює з KVM, що дозволяє йому використовувати апаратну віртуалізацію. Використовуючи ioctl, QEMU

взаємодіє з KVM для керування віртуальними машинами та взаємодії з ними. Це співпраця між QEMU та KVM дозволяє використовувати широкі можливості віртуалізації для ефективного створення та керування віртуальними середовищами. У даному режимі QEMU використовує можливості апаратної віртуалізації, якщо такі доступні на хост-системі, щоб значно покращити ефективність роботи віртуальних машин. Врна може використовувати підтримку апаратної віртуалізації таку як Intel VT-x або AMD-V, щоб прискорити виконання віртуальних машин і покращити продуктивність. QEMU керує віртуальними машинами та взаємодіє з їхніми віртуальними процесорами vCPU. QEMU використовує потоки для кожного vCPU віртуальної машини. Ці потоки плануються операційною системою, і QEMU створює враження кількох процесорів (віртуальних процесорів) для програм, які працюють всередині віртуальної машини.

Це важливо для віртуалізованих середовищ, оскільки кожен vCPU може виконувати свою роботу паралельно, надаючи враження багатопроцесорного середовища для програм, які працюють всередині віртуальної машини.

Крім того, за допомогою емуляції вводу-виводу (I/O), QEMU може підтримувати функції, які KVM може не повністю підтримувати. Це включає різноманітні сценарії взаємодії програмного забезпечення віртуальної машини з зовнішніми пристроями або середовищами, які можуть потребувати додаткової емуляції або обробки, щоб функціонувати належним чином всередині віртуального середовища.

На рис.2.5 показано потоки керування та взаємодії QEMU та KVM.

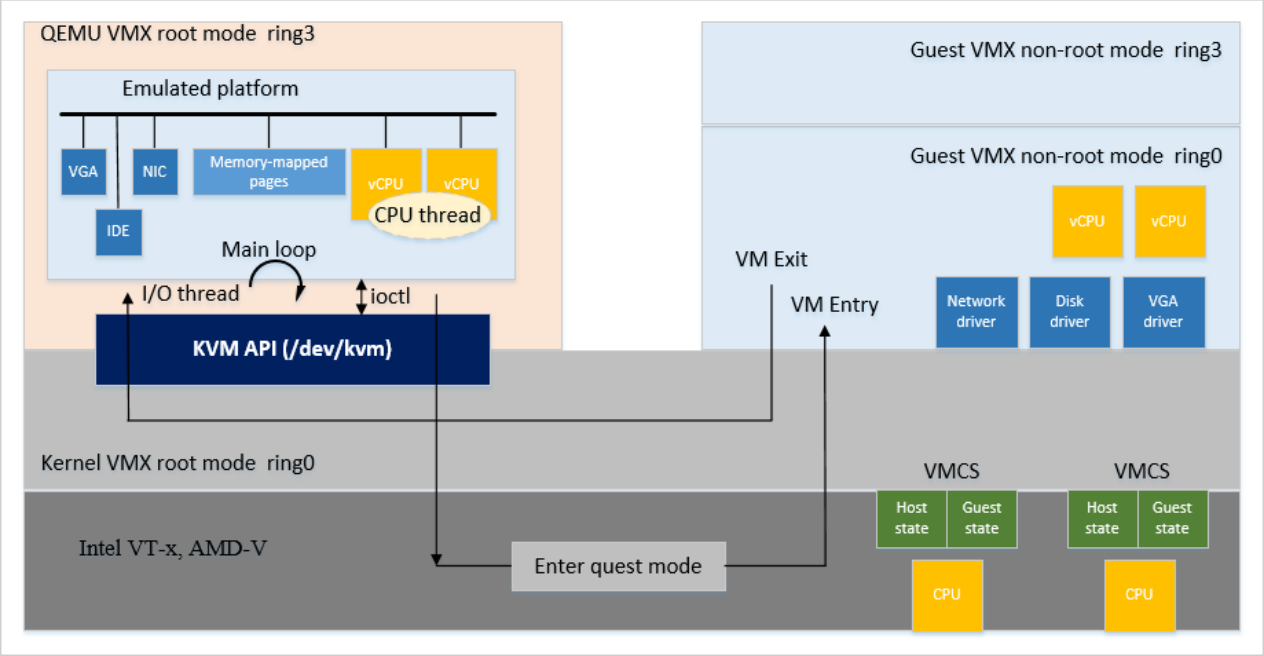


Рис. 2.5. Потоки керування та взаємодії QEMU та KVM

Загалом, QEMU - це потужний інструмент для емуляції або віртуалізації різних архітектур, який дозволяє запускати віртуальні системи на хост-платформах з іншими архітектурами.

2.4. Перевірка підтримки віртуалізації

Увімкнення віртуалізації для використання KVM зазвичай виконується через BIOS або UEFI на рівні апаратного забезпечення системи. Після активації відповідної опції потрібно перевірити, чи коректно підтримується віртуалізація в операційній системі.

В Linux можна перевірити, чи включено віртуалізацію, використовуючи команду grep -E -o '(vmx|svm)' /proc/cpuinfo | sort | uniq у вікні терміналу. Якщо результат виведення не нульовий, це означає, що віртуалізація в системі увімкнена і підтримується процесором (рис.2.6).

```
[root@oraclelinux92kvm /]# grep -E -o '(vmx|svm)' /proc/cpuinfo | sort | uniq
vmx
[root@oraclelinux92kvm /]#
```

Рис. 2.6. Вивід команди перевірка підтримки віртуалізації

Вивід показує що процесор підтримує технологію віртуалізації. Ознака vmx вказує на ввімкнену підтримку віртуалізації у процесорах Intel. Це означає, що можна використовувати KVM для віртуалізації на цій системі.

Для того щоб використовувати KVM потрібно також встановити необхідні пакети віртуалізації за допомогою пакетного менеджера.

Утиліта lscpu допомагає визначити, чи підтримує процесор технології віртуалізації, такі як Intel VT-x або AMD-V.

Ця команда виводить різноманітну інформацію про процесор, таку як архітектура, модель, кількість ядер, кеш-пам'ять, підтримку віртуалізації та інші характеристики (рис.2.7).

```
[root@oraclelinux92kvm /]# lscpu
Architecture:            x86_64
  CPU op-mode(s):        32-bit, 64-bit
  Address sizes:         45 bits physical, 48 bits virtual
  Byte Order:            Little Endian
CPU(s):                  2
  On-line CPU(s) list:   0,1
Vendor ID:               GenuineIntel
  BIOS Vendor ID:        GenuineIntel
  Model name:            Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz
    BIOS Model name:     Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz
    CPU family:          6
    Model:               158
    Thread(s) per core:  1
    Core(s) per socket:  2
    Socket(s):           1
    Stepping:            9
    BogoMIPS:            5616.00
    Flags:               fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 c
                         lflush mmx fxsr sse sse2 ss ht syscall nx pdpe1gb rdtscp lm constant_tsc
                         arch_perfmon nopl xtopology tsc_reliable nonstop_tsc cpuid tsc_known_freq
                          pni pclmulqdq vmx ssse3 fma cx16 pcid sse4_1 sse4_2 x2apic movbe popcnt
                         tsc_deadline_timer aes xsave avx f16c rdrand hypervisor lahf_lm abm 3dnow
                         prefetch cpuid_fault invpcid_single pti ssbd ibrs ibpb stibp tpr_shadow e
                         pt vpid ept_ad fsgsbase tsc_adjust bmi1 avx2 smep bmi2 invpcid rdseed adx
                          smap clflushopt xsaveopt xsavec xgetbv1 xsaves arat vnmi md_clear flush_
                         l1d arch_capabilities
Virtualization features:
  Virtualization:        VT-x
  Hypervisor vendor:     VMware
  Virtualization type:   full
```

Рис. 2.7. Вивід інформації про центральний процесор

Вивід команди lscpu показує, що процесор має підтримку віртуалізації через технологію VT-x, яка є характерною для процесорів від Intel. Також видно, що

використовується гіпервізор VMware, що підтверджує підтримку вкладеної віртуалізації.

Команда virt-host-validate у Linux призначена для перевірки того, чи готовий хост до виконання віртуалізації. Вона перевіряє наявність необхідного обладнання та конфігурації програмного забезпечення, щоб переконатися, що система здатна ефективно запускати віртуальні машини.

Дана команда перевіряє, чи процесор підтримує апаратну віртуалізацію, чи ядро Linux має необхідні модулі для віртуалізації, такі як KVM та перевіряє налаштування та стан служби libvirt, яка є ключовою для управління віртуальними машинами. Також дана команда перевіряє мережеві налаштування, які є важливими для віртуальних машин, щоб забезпечити їх підключення до мережі та переконується, що встановлено всі необхідні програми та утиліти для віртуалізації, такі як QEMU (рис.2.8).

```
[root@oraclelinux92kvm /]# virt-host-validate
  QEMU: Checking for hardware virtualization                                 : PASS
  QEMU: Checking if device /dev/kvm exists                                   : PASS
  QEMU: Checking if device /dev/kvm is accessible                            : PASS
  QEMU: Checking if device /dev/vhost-net exists                             : PASS
  QEMU: Checking if device /dev/net/tun exists                               : PASS
  QEMU: Checking for cgroup 'cpu' controller support                         : PASS
  QEMU: Checking for cgroup 'cpuacct' controller support                     : PASS
  QEMU: Checking for cgroup 'cpuset' controller support                      : PASS
  QEMU: Checking for cgroup 'memory' controller support                      : PASS
  QEMU: Checking for cgroup 'devices' controller support                     : PASS
  QEMU: Checking for cgroup 'blkio' controller support                       : PASS
  QEMU: Checking for device assignment IOMMU support                         : PASS
  QEMU: Checking if IOMMU is enabled by kernel                               : PASS
```

Рис. 2.8. Вивід команди virt-host-validate

Вивід команди virt-host-validate демонструє, що налаштування відповідають вимогам для використання віртуалізації з використанням QEMU та KVM в системі Oracle Linux. Використання цієї команди допомагає виявити та вирішити потенційні проблеми, пов'язані з віртуалізацією, перш ніж почнеться створення та управління віртуальними машинами.

2.5. Висновки до розділу

У цьому розділі було розглянуто принципи та особливості роботи гіпервізора KVM. KVM є гіпервізором у ядрі Linux, який надає підтримку для віртуалізації на платформі x86. Він використовує апаратну віртуалізацію в процесорах Intel та AMD, таку як Intel VT-x та AMD-V. KVM співпрацює з QEMU, який допомагає у створенні та управлінні віртуальними машинами. QEMU забезпечує інтерфейс для управління віртуальними середовищами та емуляцію пристроїв, а KVM використовує апаратну віртуалізацію центрального процесора для оптимізації виконання віртуальних машин. Разом вони утворюють платформу для ефективної віртуалізації та управління ресурсами в системі.

РОЗДІЛ 3
РОЗРОБКА КЛАСТЕРА ВИСОКОЇ ДОСТУПНОСТІ НА БАЗІ ORACLE LINUX

3.1 Архітектура Oracle Linux Virtualization Manager

Oracle Linux Virtualization Manager - це платформа керування віртуалізацією, яка базується на проекті з відкритим вихідним кодом під назвою oVirt [14]. Вона призначена для управління та контролю віртуалізованою інфраструктурою на базі ядра Oracle Linux.. Ця платформа надає засоби для налаштування, моніторингу та управління всіма складовими віртуального середовища, такими як сервери-хости, віртуальні машини, сховища, мережі та користувачі.

Окрім графічного інтерфейсу, OLVM також пропонує API, який дозволяє автоматизувати управління інфраструктурою KVM через програмні сценарії або інтеграцію з іншими системами керування. Це відкриває можливості для розширення функціональності та автоматизації рутинних завдань у віртуалізованому середовищі.

OLVM базується на механізмі oVirt, який є програмою Java, побудованою на платформі JBoss. Цей механізм служить як вебсервіс для керування віртуалізованою інфраструктурою, дозволяючи здійснювати централізоване управління серверами та робочими станціями. oVirt надає рішення для управління віртуальними машинами та інфраструктурою віртуалізації, що дає змогу створювати та управляти великими середовищами віртуалізації на базі KVM та Linux

Основні функції oVirt включають:

- керування хостами Oracle Linux KVM;
- створення, розгортання, запуск, зупинка, міграція та моніторинг віртуальних машин;
- створення та керування логічними мережами;

- створення та керування доменами зберігання та віртуальними дисками;
- налаштування кластерів високої доступності, хостів і віртуальних машин і керування ними;
- міграція в реальному часі віртуальних машин без їх зупинки;
- балансування навантаження між віртуальними машинами на основі використання ресурсів та політик;
- моніторинг усіх об'єктів у віртуальному середовищі, таких як віртуальні машини, хости, сховища та мережі.

Движок oVirt, що використовується в OLVM, зв’язується із службою VDSM. VDSM виступає як агент на хостах, що працюють під управлінням KVM.

Взаємодія між oVirt та VDSM відбувається через комунікаційний протокол, який дозволяє механізму керувати віртуальними машинами та виконувати різноманітні завдання на рівні хоста. Це може включати створення нових віртуальних машин, управління ресурсами хоста, налаштування мережі та зберігання, а також моніторинг та керування віртуальними образами із шаблонів.

Ця взаємодія дозволяє централізовано керувати та контролювати віртуальні машини та ресурси, розподілені по різних фізичних серверах, що працюють під управлінням KVM, за допомогою OLVM (див.рис.3.1).

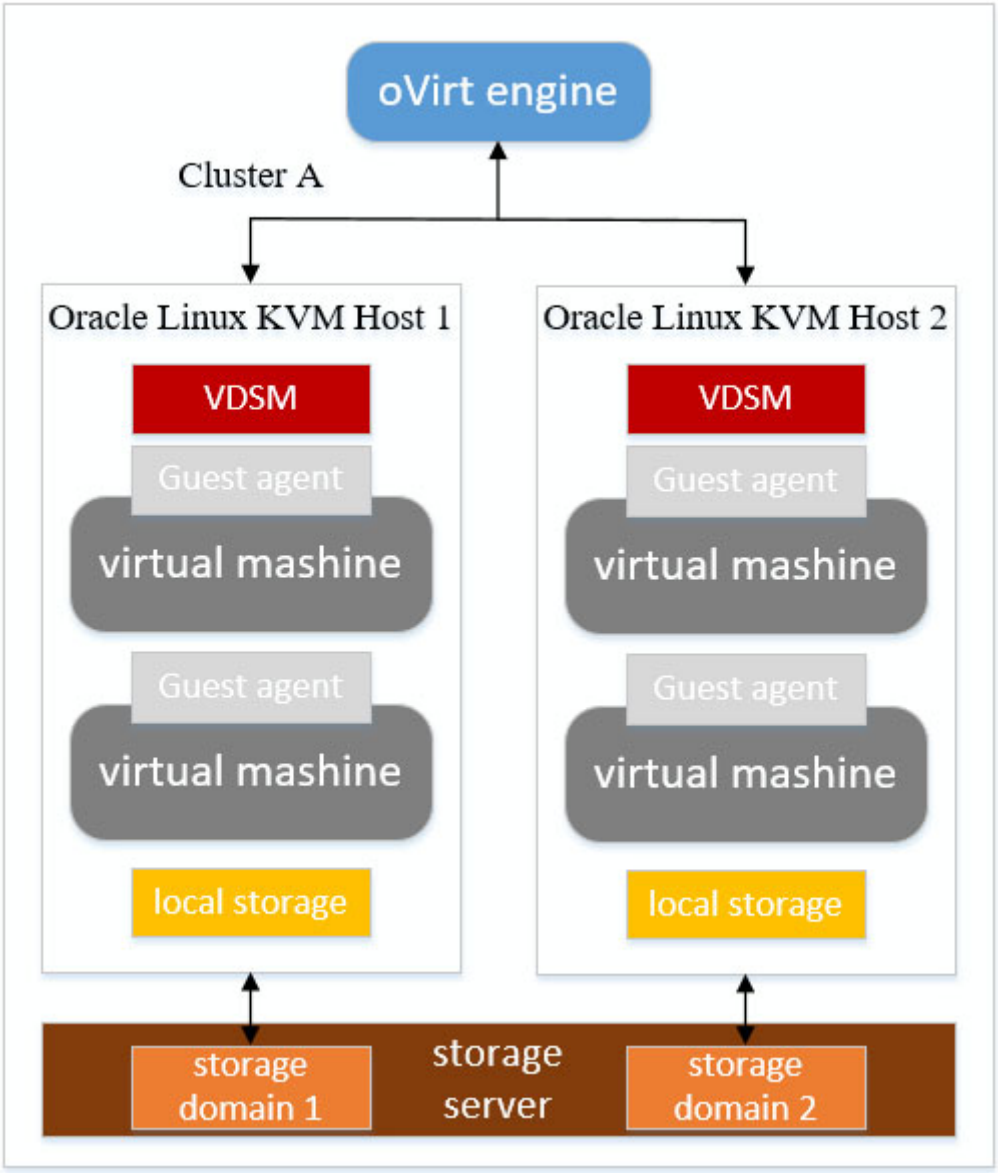


Рис. 3.1. Взаємодія oVirt та KVM

Движок oVirt є інструментом управління середовищем OLVM на сервері Oracle Linux. Хости, які працюють на платформі Oracle Linux KVM, забезпечують ресурси для функціонування віртуальних машин.

KVM, як завантажуваний модуль ядра, відповідає за наступне:

- забезпечує повну віртуалізацію, використовуючи апаратні можливості процесора;

- надає можливість хосту використовувати своє фізичне обладнання для віртуальних машин;

- працює в просторі ядра операційної системи, а віртуальні машини, що запущені на ньому, функціонують як окремі процеси QEMU у просторі користувача.

QEMU та KVM є тісно пов'язаними компонентами у віртуалізаційному стеку. QEMU виконує емуляцію апаратного забезпечення для віртуальних машин, надаючи їм віртуальні ресурси, такі як процесор, пам'ять, мережа та дискові пристрої. Крім того, KVM, який вбудований у ядро Linux, дозволяє QEMU виконувати код віртуальних машин безпосередньо на центральному процесорі, що надає операційній системі віртуальної машини безпосередній доступ до ресурсів хоста без зайвих перекладів або змін. Така взаємодія між QEMU та KVM забезпечує ефективну та продуктивну віртуалізацію.

Служба VDSM є ключовою складовою системи оркестрування віртуалізації oVirt. Вона виконує функції агента хоста, що забезпечує зв'язок та взаємодію між платформою керування oVirt та хостами, на яких запущені віртуальні машини. Служба VDSM, яка працює на хостах KVM, відповідає за управління та координацію різноманітних операцій, таких як створення, запуск, моніторинг та управління віртуальними машинами, мережами, сховищами і ресурсами хоста.

Ця служба відповідає за забезпечення оркестрування всіх дій, пов'язаних з віртуалізацією та керуванням хостами KVM, що дозволяє oVirt централізовано керувати різними аспектами інфраструктури віртуалізації.

Демон libvirt є набором програмного забезпечення для управління віртуалізацією, який забезпечує API для керування гіпервізорами, включаючи Oracle Linux KVM. libvirt може взаємодіяти з різними гіпервізорами, надаючи єдиний інтерфейс для керування різноманітними функціями віртуалізації.

Служба libvirtd працює на хостах KVM та надає API для управління віртуальними машинами та їх ресурсами. VDSM використовує libvirt для керування циклом життя віртуальних машин, такими як створення, запуск, моніторинг та зупинка віртуальних машин на хості. libvirt дозволяє збирати різні дані та статистику про віртуальні машини, що використовується для моніторингу та аналізу роботи віртуального середовища.

На рис.3.2 показана взаємодія компонентів oVirt з libvirt, VDSM, QEMU та KVM в Oracle Linux хості.

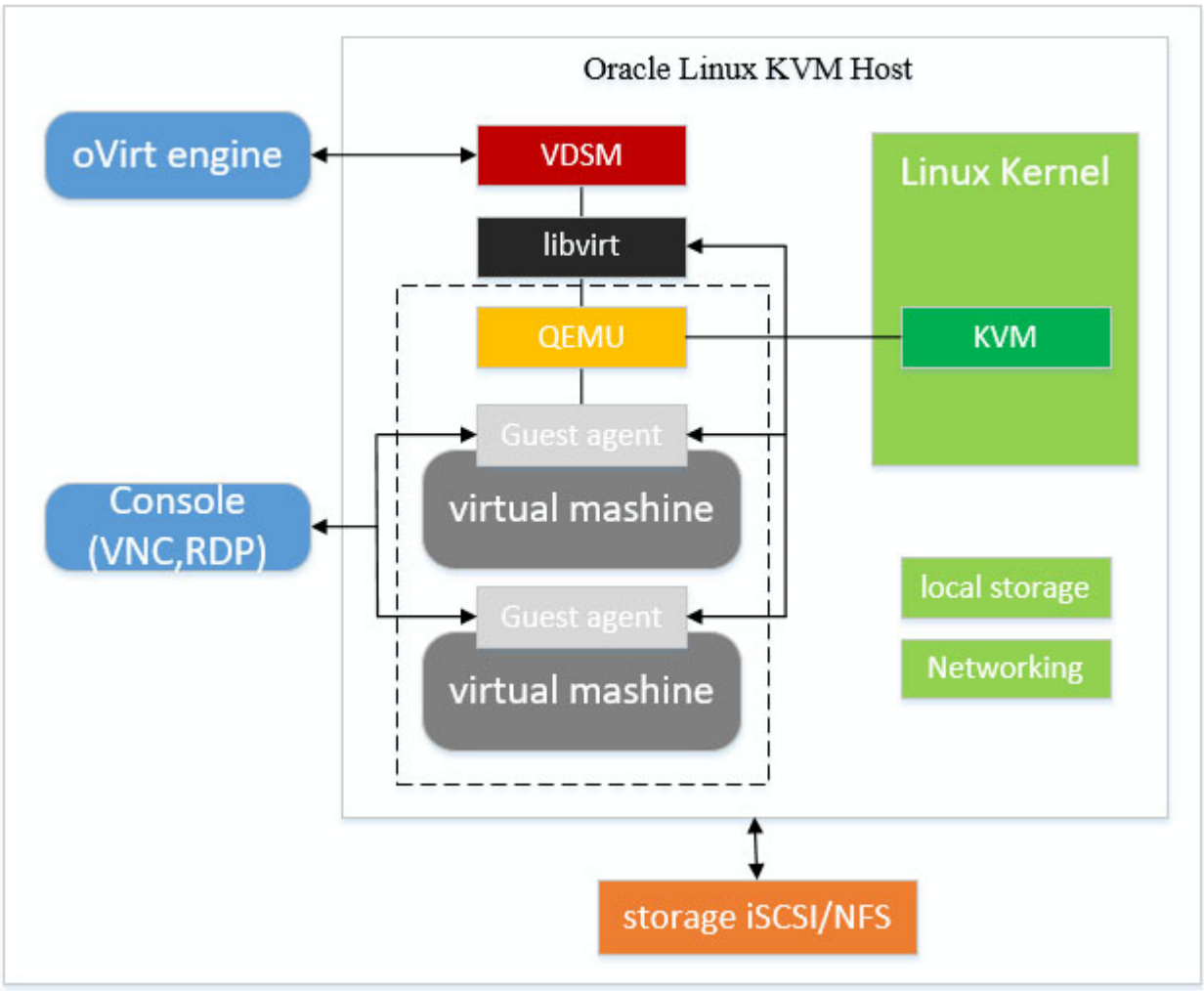


Рис. 3.2. Взаємодія oVirt з компонентами хоста Oracle Linux

Гостьовий агент є компонентом, який встановлюється всередині віртуальної машини для взаємодії з oVirt та надання інформації про використання ресурсів.

Основні функції гостьового агента включають:

- забезпечення інформацію про конфігурацію віртуальної машини, таку як ім'я, операційну систему, IP-адреси та інші параметри;

- отримання інформації про використання ресурсів (наприклад, обсяг використаної оперативної пам'яті чи навантаження процесора);

- забезпечення можливість автентифікації користувача без необхідності повторного введення автентифікаційних даних при підключенні до віртуальної машини.

Цей агент є важливим компонентом для забезпечення зв’язку та ефективної взаємодії між oVirt та гостьовими віртуальними системами.

OLVM використовує дві бази даних PostGres: engine та ovirt_engine_history.

База даних engine містить інформацію щодо стану, конфігурації та продуктивності середовища OLVM. Дані про конфігурацію та статистичні показники збираються щохвилини.

База даних ovirt_engine_history є сховищем історичної інформації про конфігурацію та статистичні показники, яку може використовувати будь-яка програма для отримання даних про центри обробки даних, кластери та хости.

OLVM має три вебпортали для управління віртуалізацією: Administration Portal для налаштування, VM Portal для управління віртуальними машинами і Monitoring Portal для моніторингу середовища.

Адміністративний портал є інтерфейсом управління oVirt engine, який надає адміністраторам можливість контролювати, створювати та підтримувати всі елементи віртуалізованого середовища через веббраузер. Деякі завдання, які можна виконувати з адміністративного порталу, включають:

- конфігурація та управління віртуальними машинами;
- налаштування мереж і сховищ даних;
- створення та керування кластерами віртуалізації;
- моніторинг ресурсів та продуктивності віртуальних середовищ;
- управління різними аспектами безпеки та доступу;
- інтеграція з різноманітними інструментами та ресурсами віртуалізації.

Портал віртуальних машин надає користувачам повне уявлення про їхні віртуальні машини і дозволяє виконувати дії, такі як запуск, зупинка, редагування та перегляд деталей віртуальної машини. Доступні дії для користувача на порталі віртуальних машин можуть бути встановлені системним

адміністратором, який може делегувати додаткові завдання адміністрування користувачам, такі як:

- створення, редагування та видалення віртуальних машин;
- управління віртуальними дисками та мережевими інтерфейсами;
- створення та використання знімків для відновлення віртуальних машин до попередніх станів.

Портал моніторингу відкриває інтерфейс Grafana, де доступні вбудовані інформаційні панелі Grafana, такі як Executive, Inventory, Service Level і Trend. Користувач може створювати власні інформаційні панелі або копіювати та налаштовувати наявні залежно від власних потреб у звітності. Інтеграція Grafana вмикається та встановлюється за замовчуванням під час налаштування двигуна системи.

Пряме підключення до віртуальних машин можна виконати за допомогою клієнтів SPICE або VNC. Обидва ці протоколи надають користувачам середовище, подібне до того, як якщо б вони використовували локально підключений монітор до системного блоку. Адміністратор може вказати протокол, який буде використовуватися для підключення до віртуальної машини під час її створення.

При використанні протоколу VNC можна відкривати консоль віртуальної машини за допомогою програми Remote Viewer або клієнта VNC. У випадку, якщо виникає потреба використовувати консольні клієнти на основі браузера, необхідно імпортувати центр сертифікації в браузер, оскільки зв'язок буде захищеним. Це може знадобитися для правильної аутентифікації та забезпечення безпеки під час взаємодії з консоллю віртуальних машин.

Також можна використати Remote Desktop Protocol, для підключення до віртуальних машин із операційною системою Windows

3.2 Планування та розгортання кластера високої доступності

Планування та розгортання кластера високої доступності є ключовим етапом у створенні стійкої та надійної інфраструктури. Цей процес включає в себе кілька кроків. На рис.3.3 показано схема тестового середовища для створення кластера високої доступності.

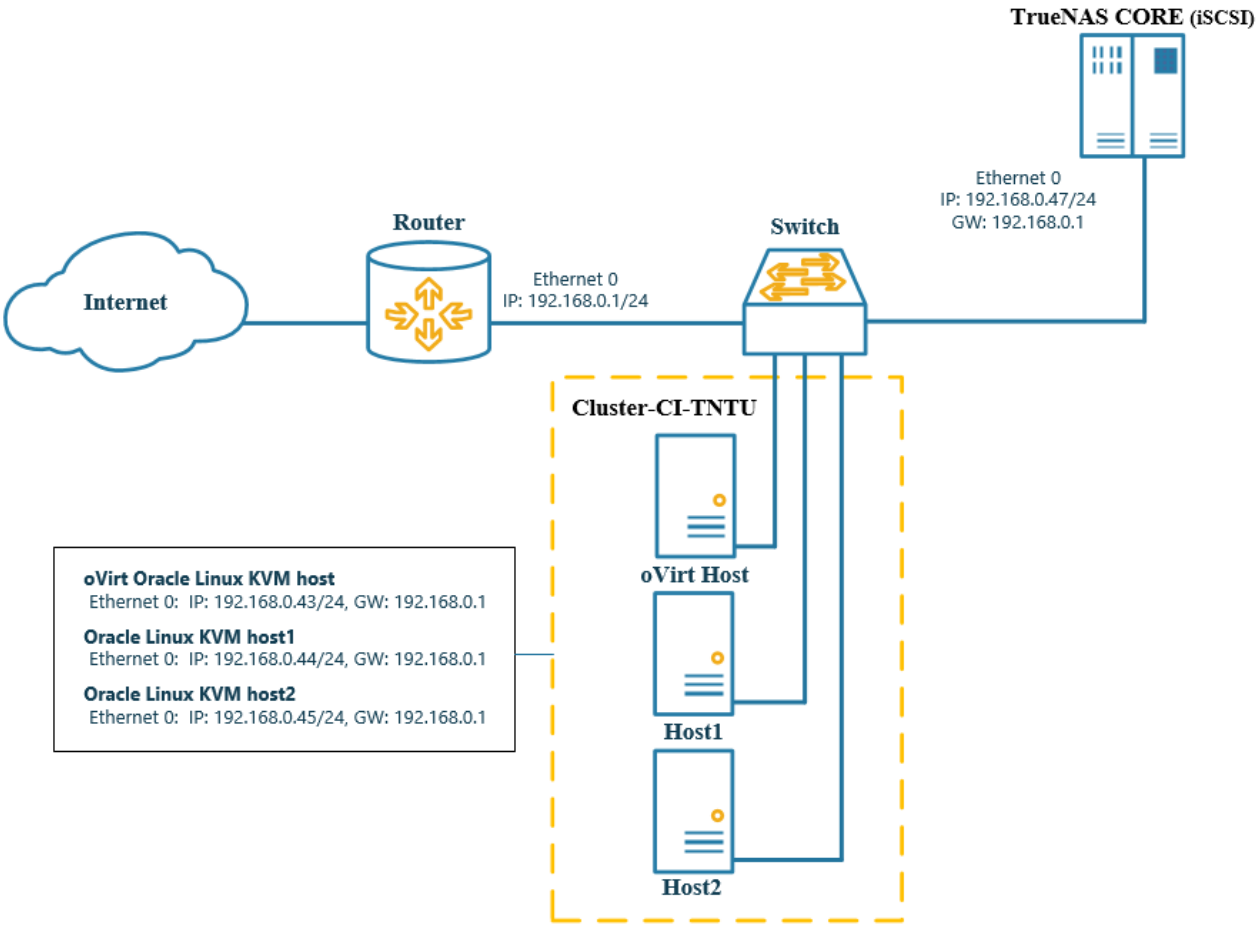


Рис. 3.3. Схема тестового середовища для створення кластера високої доступності

Ця схема є планом тестового середовища для створення кластера високої доступності. На схемі зображено з'єднання між різними компонентами, такими як маршрутизатори, комутатори, хости та сховища даних.

Інтернет підключений до маршрутизатора з IP-адресою 192.168.0.1/24. Маршрутизатор, в свою чергу, підключений до комутатора. Також на схемі

зображено кластер високої доступності під назвою Cluster-CI-TNTU з трьома хостами на базі Oracle Linux KVM з IP-адресами 192.168.0.43/24, 192.168.0.44/24 та 192.168.0.45/24 відповідно. На одному з хостів встановлено oVirt engine що є основою платформи керування віртуалізацією OLVM. Як спільне сховище для кластера використовується TrueNAS CORE з IP-адресою 192.168.0.47/24. Протокол iSCSI використовується для надання спільного сховища хостам KVM.

3.2.1. Центр обробки даних. Центр обробки даних - це узагальнена структура для управління ресурсами високого рівня в середовищі віртуалізації. Управління ресурсами на цьому рівні забезпечує логічне групування фізичних і логічних ресурсів (таких як сервери, сховища, мережі тощо), що використовуються для обробки даних в межах одного центру або групи даних.

Може існувати кілька центрів обробки даних, кожен з яких може включати свої власні кластери, хости і зберігання. Ці центри обробки даних можуть бути керовані через єдиний інтерфейс адміністрування, що значно спрощує управління та контроль за різними об'єктами віртуалізації у різних локаціях чи середовищах.

Щоб ініціалізувати центр обробки даних потрібно щоб були додані різні об'єкти, такі як кластер, хости і зберігання (сховища даних). Це дозволяє адміністраторові чітко визначити та організувати ресурси для подальшого використання в рамках віртуального середовища.

На рис.3.4 показано три хости KVM, які розміщені в центрі обробки даних DataCenter-CI-TNTU в кластері Cluster-CI-TNTU.

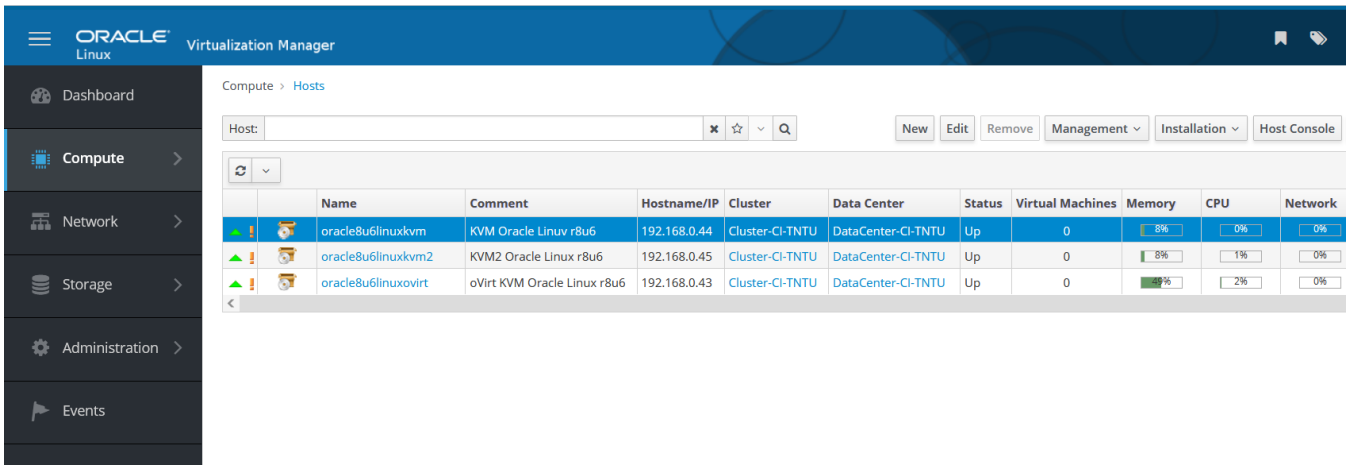


Рис. 3.4. Хости Oracle Linux KVM в кластері

На одному з хостів встановлено oVirt Engine. Окрім основного призначення даний хост є резервним KVM хостом.

3.2.2. Кластер. Кластер - це група хостів, які працюють разом із певними політиками та параметрами налаштування. У віртуальному середовищі Oracle Linux KVM кластери відіграють важливу роль в управлінні віртуальними машинами.

Кластер складається з групи логічно об'єднаних хостів Oracle Linux KVM. Ці хости працюють у визначеному спільною політикою і налаштуваннями середовищі, такими як ресурси, доступність і спільний доступ до сховищ даних. Хости KVM у межах кластера повинні мати схожий тип процесора (Intel або AMD) і відповідати певним умовам працездатності, щоб гарантувати сумісність і рівномірність роботи віртуальних машин. Віртуальні машини можуть динамічно призначатися будь-якому хосту KVM у кластері, згідно з політиками кластера та налаштуваннями віртуальних машин. Це дає можливість працювати віртуальним машинам навіть у випадку, якщо один або декілька хостів недоступні. Завдяки відсутності прив’язки віртуальних машин до конкретного хосту, вони можуть переноситися між хостами у кластері відповідно до налаштувань та політик розподілу навантаження. На рівні кластера визначаються політики та параметри щодо розподілу ресурсів, таких як обсяги

пам’яті, обчислювальна потужність, а також політики відновлення в разі виникнення помилок або відмов.

Кластери створюють зручну та надійну структуру для керування віртуальними машинами та ресурсами у віртуальному середовищі Oracle Linux KVM, забезпечуючи більш високу доступність та ефективне використання обчислювальних ресурсів.

На рис.3.5 показано загальні характеристики кластера Cluster-CI-TNTU.

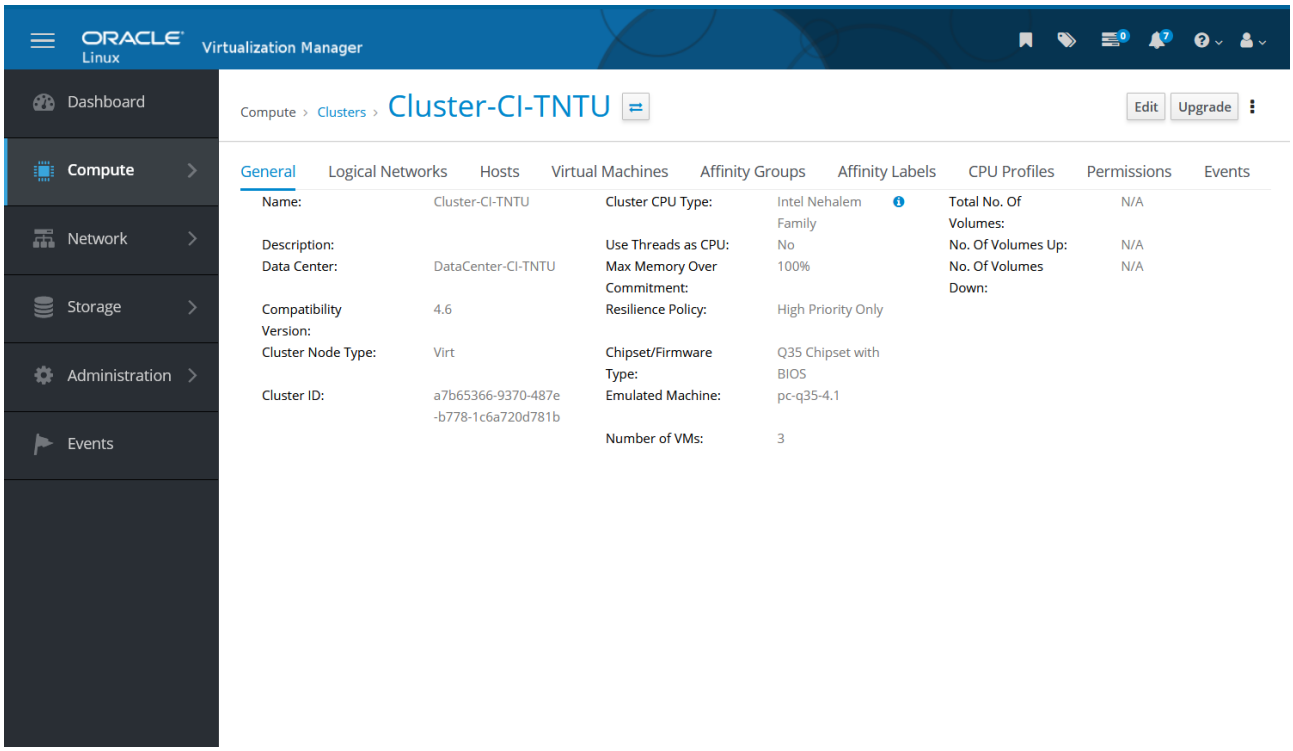


Рис. 3.5. Характеристики кластера Cluster-CI-TNTU

3.2.3. Хости Oracle Linux KVM. OLVM є рішенням для управління віртуалізованим середовищем, що ґрунтується на гіпервізорі KVM. Для коректної роботи даної платформи потрібно встановити Oracle Linux на фізичний сервер з ядром Unbreakable Enterprise Kernel для використання як гіпервізор KVM.

OLVM дозволяє керувати багатьма хостами Oracle Linux KVM, кожен з яких може запускати кілька віртуальних машин одночасно. Віртуальні машини працюють як окремі процеси та потоки Linux на хості KVM. В середовищі

OLVM роль oVirt хосту може бути також призначена для хоста, на якому працює Oracle Linux без KVM.

На рис.3.6 показано загальні характеристики хоста KVM oracle8u6linuxkvm.

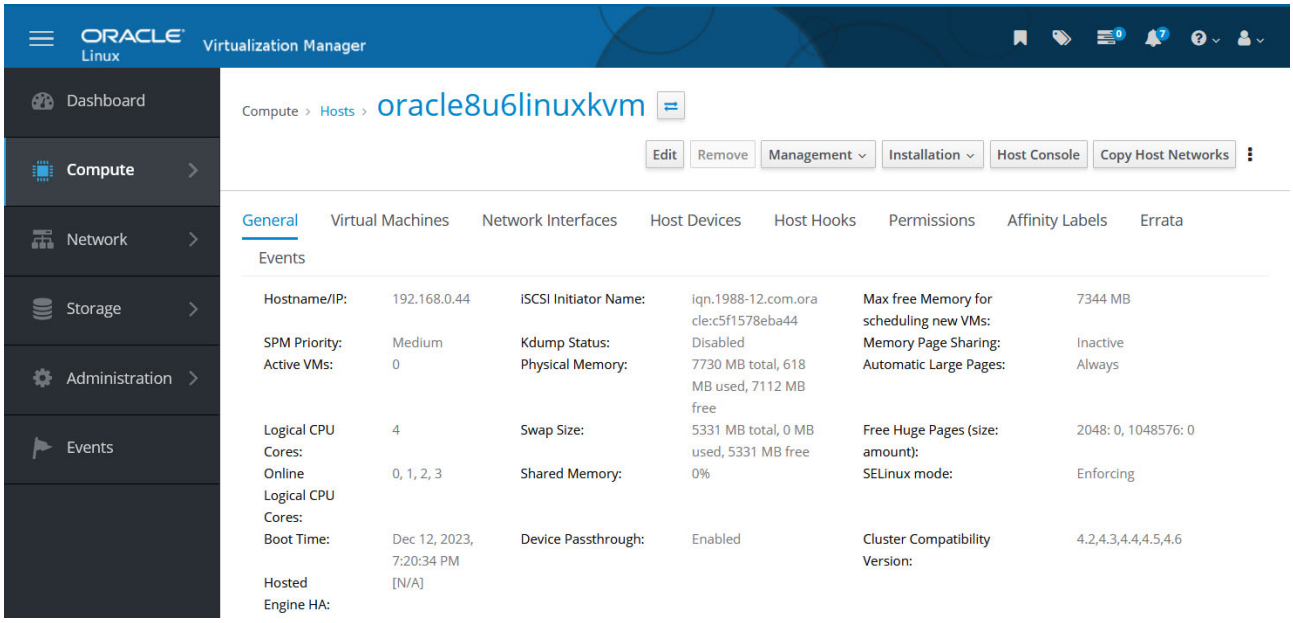


Рис. 3.6. Характеристики хоста oracle8u6linuxkvm

3.2.4. Віртуальні машини. За допомогою OLVM можна створювати віртуальні машини з певною конфігурацією або клонувати їх з існуючих шаблонів в пулах віртуальних машин. Також є можливість імпортувати файли OVA з будь-якого хоста у середовище в центрі обробки даних. Гостьові агенти та драйвери надають віртуальним машинам функції відстеження використання ресурсів та можливість вимикати та перезавантажувати їх з порталу адміністрування.

Шаблон - це копія віртуальної машини, яка використовується для спрощення створення подібних віртуальних машин у майбутньому. Шаблони фіксують конфігурацію програмного та апаратного забезпечення, встановленого на віртуальній машині, на якій базується шаблон. Віртуальні машини, створені на основі шаблону, мають той самий тип мережевої карти та драйвер, що й вихідна віртуальна машина, але отримують унікальні MAC-адреси.

Знімок - це зображення стану віртуальної машини в конкретний момент часу. Використовується для відновлення віртуальної машини до певного стану, якщо це буде потрібно, і цей механізм не слід використовувати як основний процес резервного копіювання.

На рис.3.7 показано загальні характеристики віртуальної машини Mikrotik RouterOS CHR. Це операційна система, яка використовується для роботи як маршрутизатор на віртуальних платформах, таких як VMware, Hyper-V, KVM, XEN та може бути встановлена в хмарних середовищах.

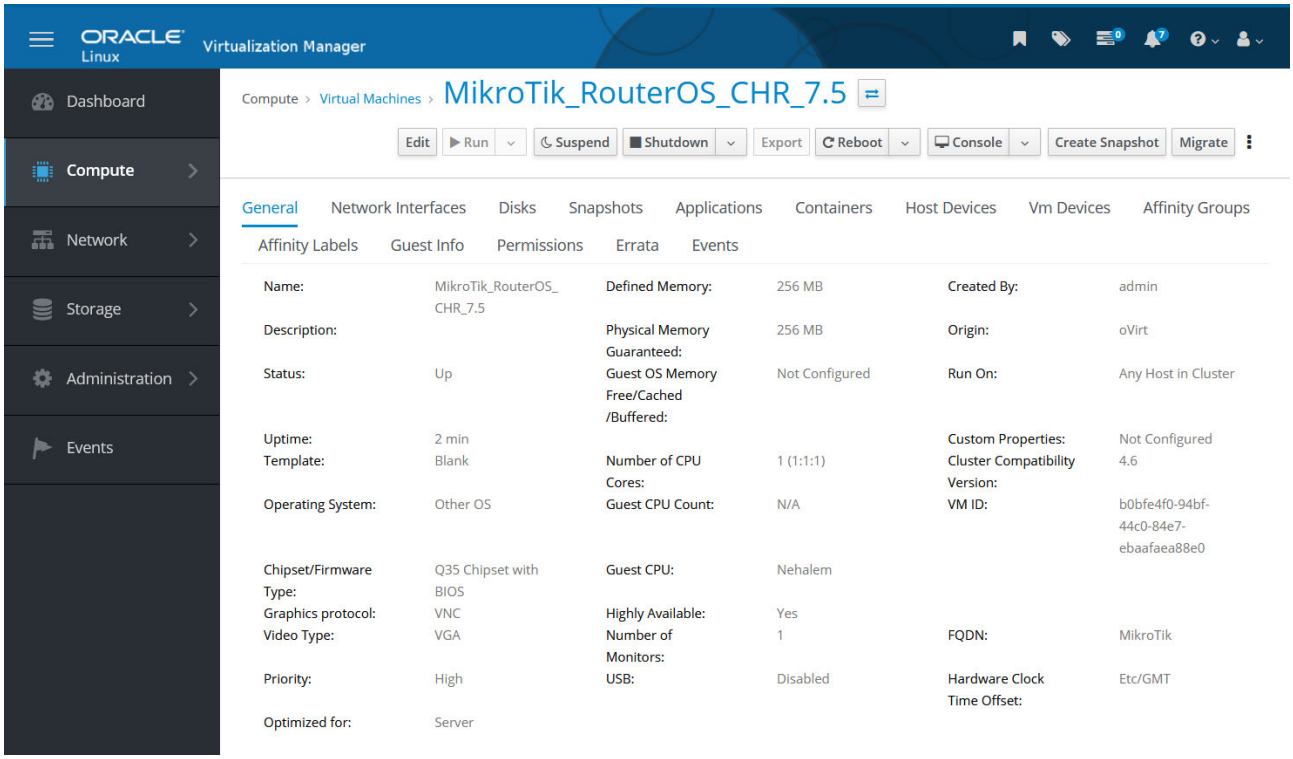


Рис. 3.7. Характеристики віртуальної машини

3.2.5. Конфігурація мережі. Рекомендується, щоб хост OLVM і всі хости Oracle Linux KVM мали повне доменне ім'я (FQDN). Це допомагає забезпечити правильне ідентифікування інших машин у мережі.

Oracle рекомендує використовувати службу DNS для розпізнавання імен. Це означає, що імена машин повинні бути зареєстровані у DNS-сервері, щоб інші машини у мережі могли їх ідентифікувати за допомогою DNS-запитів.

Це є більш надійним методом, оскільки імена машин можуть бути автоматично розпізнані і змінюватися через DNS без необхідності вручного оновлення на кожній машині.

Хоча можна використовувати файл /etc/hosts для розпізнавання імен, це не є найбільш оптимальним методом через можливість помилок і додаткову роботу, яку потрібно виконати для оновлення файлу на кожному хості.

Окрім того, використання /etc/hosts може бути менш масштабованим та складнішим у випадку розширення мережі або зміни імен машин.

Служби DNS, які використовуються для розпізнавання імен, мають бути розміщені за межами середовища OLVM. Це означає, що DNS-сервери повинні бути доступні за межами віртуальної мережі, щоб забезпечити правильне розпізнавання імен хостів.

В OLVM можна налаштувати логічні мережі для призначення ресурсів, необхідних для забезпечення підключення до мережі хостів Oracle Linux KVM. Однією з таких логічних мереж є мережа керування, яка призначена для управління мережевими інтерфейсами хостів.

Спочатку визначається логічна мережа для центру обробки даних, яка потім застосовується до одного або кількох кластерів. Після цього проводиться налаштування хостів шляхом призначення логічних мереж фізичним інтерфейсам хостів (див. рисунок 3.8).

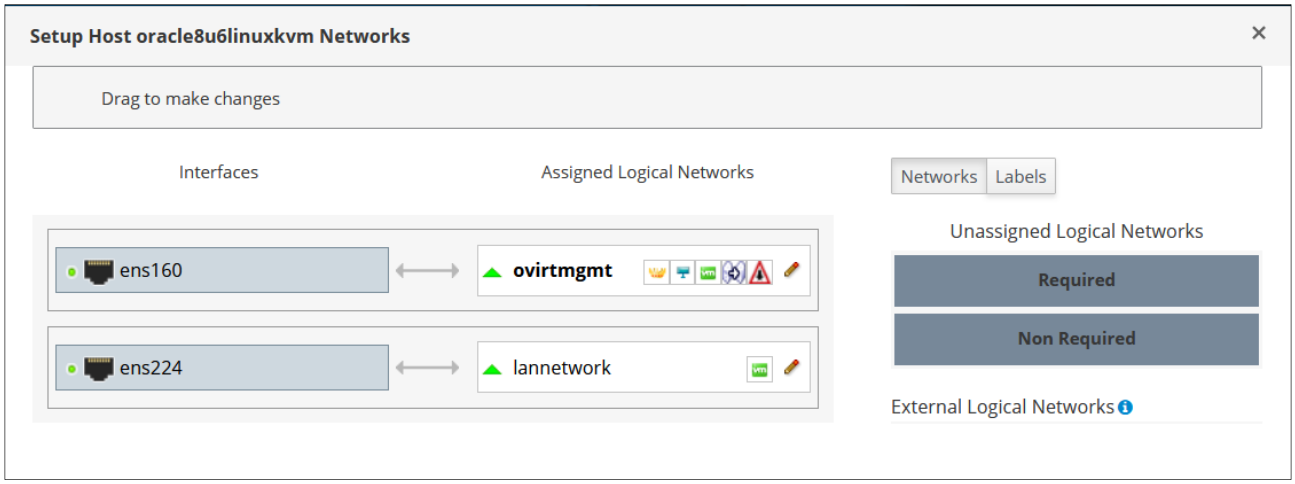


Рис. 3.8. Логічні та фізичні мережі хоста

Після застосування мережі на всіх хостах у кластері мережа активується та починає працювати. Усі ці дії виконуються через портал адміністрування.

На рівні кластера можна налаштувати різні мережеві ролі для логічних мереж з метою визначення їх призначення.

Мережа керування (management network) - це мережа, яка використовується для зв'язку між OLVM і хостами. Вона дозволяє керувати та адмініструвати хости, управляти ресурсами та конфігурацією системи віртуалізації.

Мережа віртуальної машини (VM network) - це мережа, яка використовується для зв'язку віртуальних машин між собою та зовнішнім середовищем. Віртуальні машини підключаються до цієї мережі через віртуальні мережеві адаптери для обміну даними та комунікації з іншими пристроями у мережі.

Дисплейна мережа (display network)- це мережа, яка використовується для підключення клієнтів до графічних консолей віртуальних машин через протоколи, такі як VNC або RDP. Вона дозволяє користувачам взаємодіяти з графічним інтерфейсом віртуальних машин.

Мережа міграції (migration network) - це мережа, яка використовується для міграції віртуальних машин між різними хостами у кластері. Вона забезпечує зручність та ефективність переміщення віртуальних машин без втрати продуктивності чи доступності.

Ці мережеві ролі дозволяють ефективно розподіляти та керувати трафіком у віртуалізованому середовищі, надаючи різні функціональні можливості для керування, комунікації та оптимізації роботи віртуальних машин у кластері.

Основна логічна мережа (ovirtmgmt) створюється за замовчуванням і використовується для всіх мережевих зв'язків у центрі обробки даних. Проте, для кращого управління та розділення мережевого трафіку рекомендується налаштувати додаткові логічні мережі згідно зі своїми потребами. Одна з логічних мереж може бути налаштована як маршрут за замовчуванням для хостів. Це важливо для правильної маршрутизації мережевого трафіку. Хости KVM підключаються до логічних мереж, які не є мережами віртуальних машин,

безпосередньо за допомогою фізичних мережевих інтерфейсів або інтерфейсів VLAN. Для мереж віртуальних машин створюються мости на хості для кожної логічної мережі. Віртуальні мережеві адаптери (vNIC) віртуальних машин підключаються до цих мостів за необхідності. Це дозволяє віртуальним машинам з'єднуватися з мережею через фізичні мережеві інтерфейси або інтерфейси VLAN.

Цей підхід до управління мережами дозволяє створювати, налаштовувати та управляти різними мережами віртуалізованого середовища, щоб забезпечити ефективність та безпеку мережевого трафіку у центрі обробки даних.

На рис.3.9 показано мережеві налаштування віртуальної машини Mikrotik RouterOS CHR.

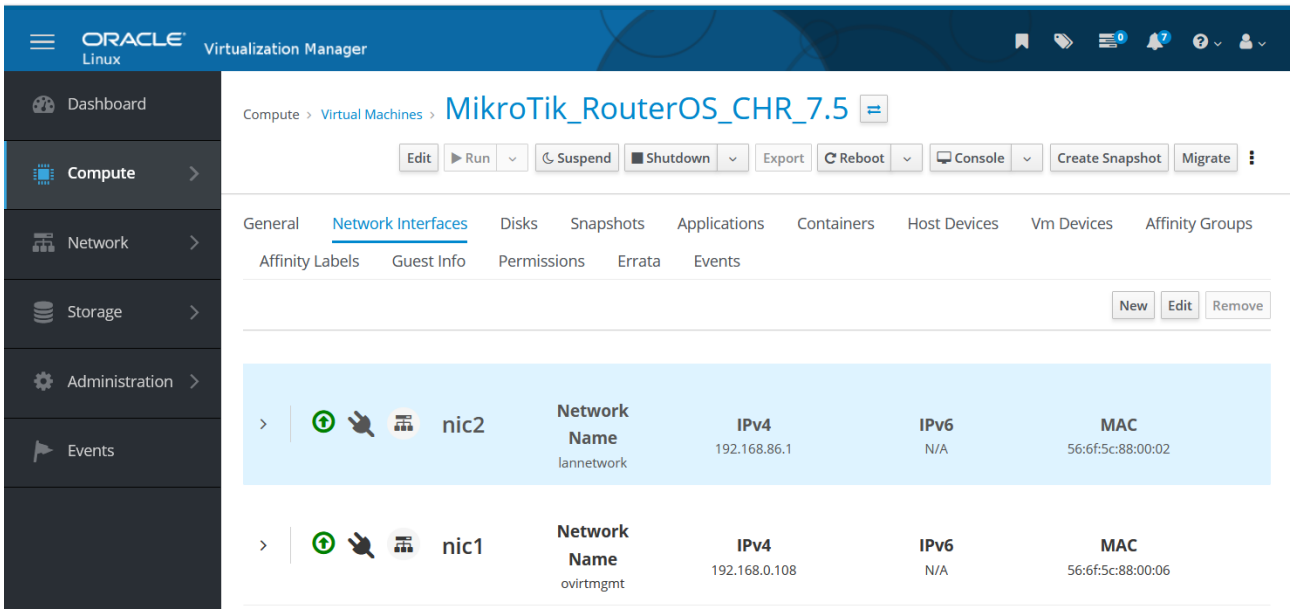


Рис. 3.9. Логічні мережеві інтерфейси віртуальної машини

Віртуальна машина використовує контролер віртуального мережевого інтерфейсу (vNIC), який є віртуальним пристроєм для підключення до логічної мережі. Цей віртуальний інтерфейс дозволяє віртуальній машині спілкуватися із мережею.

Міст (bridge) - це програмний мережевий пристрій на хості KVM, який функціонує як комутатор для об'єднання та пересилання мережевого трафіку між

фізичними та віртуальними мережевими інтерфейсами. vNIC завжди підключаються до мосту на хості KVM, що дозволяє їм використовувати фізичне мережеве підключення. Міст дозволяє vNIC віртуальних машин спільно використовувати фізичне мережеве підключення. Це дозволяє віртуальним машинам ефективно спілкуватися з іншими пристроями та ресурсами у мережі через мережеві інтерфейси.

OLVM автоматично надає MAC-адреси для кожного віртуального мережевого інтерфейсу (vNIC). Ці адреси вибираються з попередньо визначеного діапазону призначеного для кластера. Кожна MAC-адреса унікальна, що дозволяє відповідному vNIC бути ідентифікованим в мережі. vNIC забезпечують підключення віртуальних машин до логічної мережі. Кожен vNIC може мати свою IP-адресу, яку можна призначити за допомогою DHCP або встановити статично через інструменти операційної системи віртуальної машини. Для використання DHCP необхідно налаштувати DHCP-сервер у логічній мережі. Віртуальні машини можуть спілкуватися між собою в рамках віртуальної мережі та, в залежності від конфігурації логічної мережі, взаємодіяти з глобальними мережами, такими як Інтернет.

3.2.6. Засоби зберігання. Oracle Linux Virtualization Manager використовує централізовану систему зберігання для управління образами дисків віртуальних машин, ISO-файлами та знімками.

Ця система зберігання може бути сконфігурована за допомогою різних технологій, таких як: NFS, iSCSI, протокол Fibre Channel, Gluster FS. Також є можливість налаштування локального сховища, яке прямо підключене до хостів. Це дозволяє зберігати дані прямо на локальних дисках хостів, що може бути корисним у випадках, коли потрібна велика швидкість доступу до даних без залучення мережевих ресурсів

У центрі обробки даних ініціалізація не може бути завершена, якщо відсутній або не активований домен зберігання. Для правильної роботи системи, сховище повинне розташовуватися в тій самій підмережі, що й хости Oracle

Linux KVM, які будуть використовувати це сховище, щоб уникнути можливих проблем з маршрутизацією.

Домен зберігання представляє собою набір образів. Цей домен містить повні образи віртуальних машин, знімки віртуальних машин або файли ISO, які використовуються для віртуалізації (див.рис.3.10). Це є основою для управління та зберіганням даних, які використовуються віртуальними середовищами.

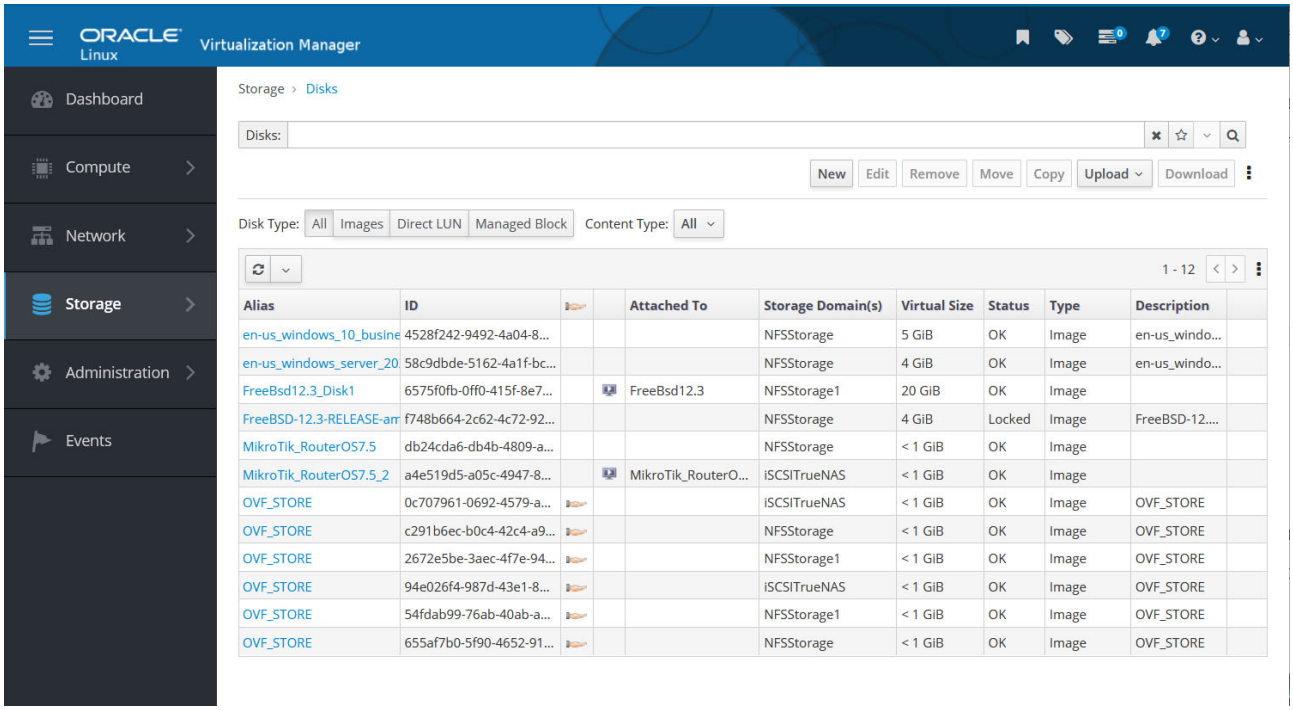


Рис. 3.10. Вміст домену зберігання

OLVM підтримує домени зберігання, які є блоковими пристроями (SAN - iSCSI або FCP) або файловою системою (NAS - NFS або Gluster) (рис.3.11).

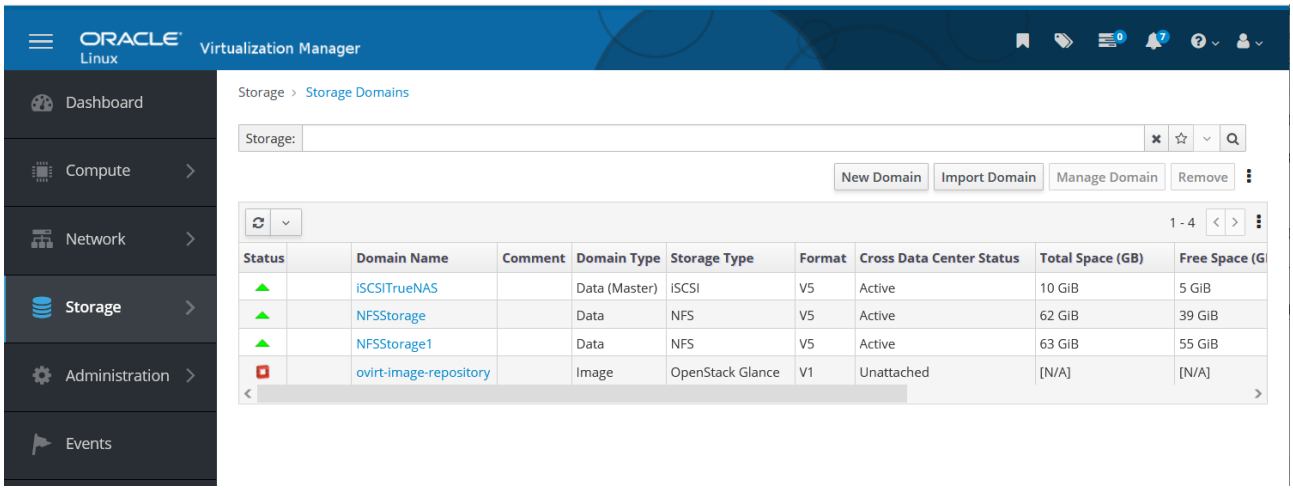


Рис. 3.11. Налаштовані домени зберігання

В системах зберігання, таких як NFS та Gluster, усі віртуальні диски, шаблони та знімки віртуальних машин зберігаються у вигляді файлів на файловій системі у відповідному розташуванні.

В SAN, такому як iSCSI або FCP, кожен віртуальний диск, шаблон або знімок віртуальної машини може бути представлений як логічний том. Тобто вони зберігаються як блочні пристрої або логічні томи, які надаються та управляються через протоколи передачі даних по мережі (наприклад iSCSI) або через магістраль зв'язку, таку як Fibre Channel.

Віртуальні машини, які використовують один і той самий домен зберігання, можуть бути переміщені між хостами, які належать до того самого кластера. Це дозволяє реалізувати функцію міграції віртуальних машин між різними фізичними серверами у віртуалізованому середовищі для оптимізації ресурсів та підвищення надійності системи.

Кожен центр обробки даних повинен мати принаймні один домен зберігання даних. Домени даних не можуть бути спільно використані між різними центрами обробки даних. Це обмеження обумовлене потребою управління та зберіганням даних на рівні конкретного центру обробки, що дозволяє краще контролювати доступ та безпеку інформації.

Від’єднання домену зберігання від конкретного центру обробки даних припиняє активну асоціацію між цим доменом та цим центром обробки даних.

Однак це від'єднання не призводить до автоматичного видалення домену зберігання з середовища. Після від’єднання домен зберігання все ще існує у віртуальній інфраструктурі, і дані, такі як віртуальні машини та шаблони, залишаються пов'язаними з цим відокремленим доменом зберігання.

Такий відокремлений домен зберігання можна повторно приєднати до іншого центру обробки даних у майбутньому, і дані, які залишилися у цьому домені зберігання, можна буде використати після приєднання.

Диспетчер пулу сховищ SPM у системі віртуалізації OLVM, представляє собою роль керування, що призначається одному з хостів у центрі обробки даних. Основне завдання SPM - це керування та управління доменами зберігання в центрі обробки даних. SPM відповідає за координацію метаданих в доменах сховища, які зазвичай містять віртуальні диски (зображення), знімки та шаблони. Крім цього, SPM керує створенням, видаленням та управлінням цими ресурсами, в тому числі виділенням пам'яті для розподілених блокових пристроїв у системах зі зберіганням SAN. SPM здійсеює контроль доступу до сховища та забезпечення цілісності метаданих в доменах зберігання. Він відповідає за управління та сприяння процесам, пов'язаним із зберіганням віртуальних образів, їх створенням, зміною, та видаленням, а також за забезпеченням відведення необхідних ресурсів для їх функціонування в системі зберігання.

Роль диспетчера пулу сховищ може бути призначена для будь-якого хоста у центрі обробки даних, і цей хост продовжує розміщувати віртуальні ресурси навіть в той час, коли він виконує роль SPM.

Встановлення пріоритетів для ролі SPM для різних хостів дає можливість визначити, якому саме хосту буде призначено завдання SPM. Оскільки виконання ролі SPM потребує певних ресурсів з хоста, важливо правильно налаштувати пріоритети хостів, щоб система могла призначати цю роль на базі цих пріоритетів. Оскільки надійність SPM є критичною, система визначає нового SPM, якщо поточний хост, який обслуговує SPM, стає недоступним. Це означає, що якщо поточний хост, який виконує роль SPM, стає недоступним, то хосту з вищим пріоритетом SPM буде автоматично призначено роль SPM перед хостом

із нижчим пріоритетом SPM. Таким чином, система забезпечує постійну доступність SPM, навіть у випадку відмови поточного хоста.

Якщо для зберігання використовується NFS або локальне сховище, SPM автоматично створює тонкий віртуальний диск за замовчуванням. Тонкий віртуальний диск - це механізм, коли простір на сховищі не виділяється наперед, а створюється динамічно за потребою, що дозволяє ефективніше використовувати місце.

У випадку використання сховища iSCSI або інших блокових пристроїв, номери логічних пристроїв (LUN) призначаються SPM. Після цього SPM створює групу томів поверх LUN та логічних томів, які використовуються для створення віртуальних дисків для віртуальних машин. При цьому SPM автоматично виділяє певний простір за замовчуванням, щоб забезпечити використання цих ресурсів для нових віртуальних дисків.

Локальне сховище означає сховище, яке фізично прив'язане безпосередньо до окремого хоста Oracle Linux KVM. Це може бути локальний фізичний диск.

Коли хост KVM використовує локальне сховище, він автоматично приєднується до окремого кластера, в якому він єдиний учасник. Це виникає з того, що в кластерах, що складаються з декількох хостів, потрібно спільне сховище, до якого мають доступ всі хости для обміну даними і ресурсами.

Коли використовується локальне сховище, деякі розширені можливості, такі як жива міграція (live migration), планування ресурсів є недоступні. Це через обмежену можливість обміну даними та ресурсами між окремим хостом і іншими частинами кластера.

Для створення кластера високої доступності на базі Oracle Linux було використано TrueNAS CORE, як сховище даних [15].

TrueNAS CORE надає засоби для створення масштабованих сховищ даних. Це безкоштовна і відкрита система для зберігання даних, яка базується на операційній системі FreeBSD. Вона надає багатофункціональні можливості для зберігання та управління даними на серверах зберігання, включаючи функції безпеки, масштабованості та надійності.

Основна функція полягає у створенні мережевих сховищ з використанням протоколів мережевого сховища, таких як SMB, NFS, Apple Filing Protocol (AFP), iSCSI та використання технологій ZFS для забезпечення надійного та ефективного зберігання даних. iSCSI - це протокол, який дозволяє використовувати блочне сховище через IP-мережу. Протокол дозволяє надати доступ до збережених даних як блоків на віддаленому сервері і використовувати їх на віддаленому клієнтському пристрої так, ніби це локальний диск. NFS - це мережевий протокол файлової системи, який дозволяє робочим станціям та серверам звертатися до файлів і даних на віддалених комп'ютерах так, ніби вони знаходяться в локальній системі. Широко використовується в середовищах Unix та Linux.

На рис.3.12 показано налаштування пулу зберігання в TrueNAS CORE для використання по протоколу iSCSI в OLVM.

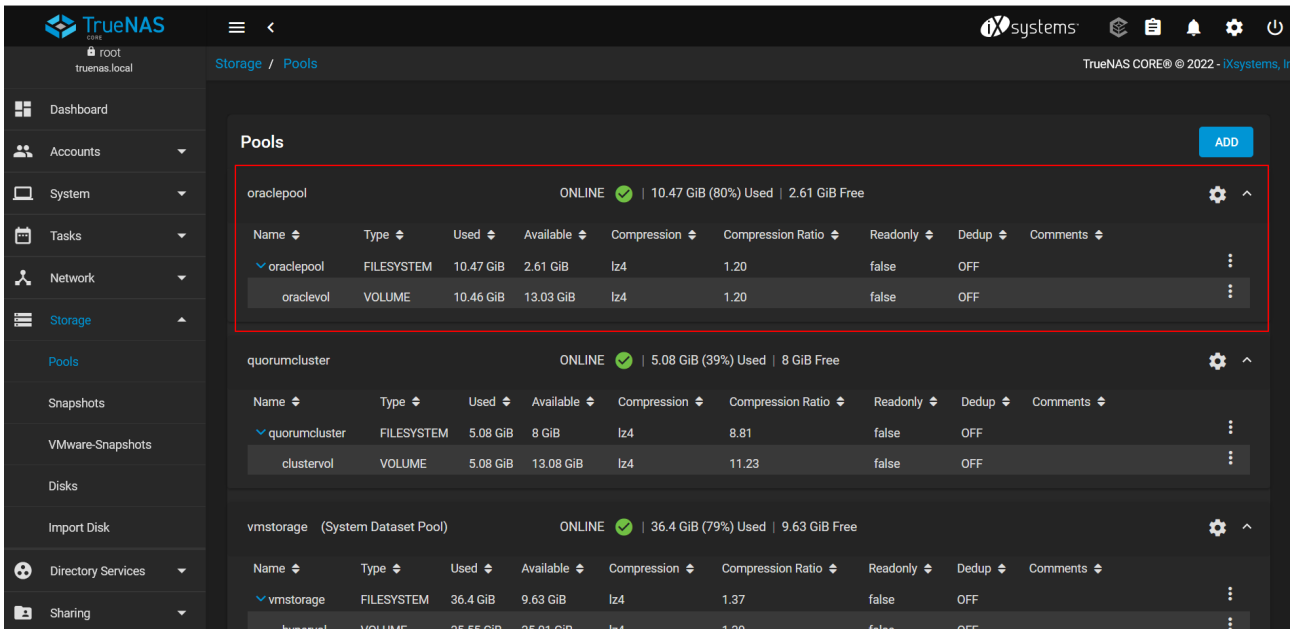


Рис. 3.12. Пули зберігання в TrueNAS CORE

На рис.3.13 показано під’єднання сховища iSCSI в OLVM.

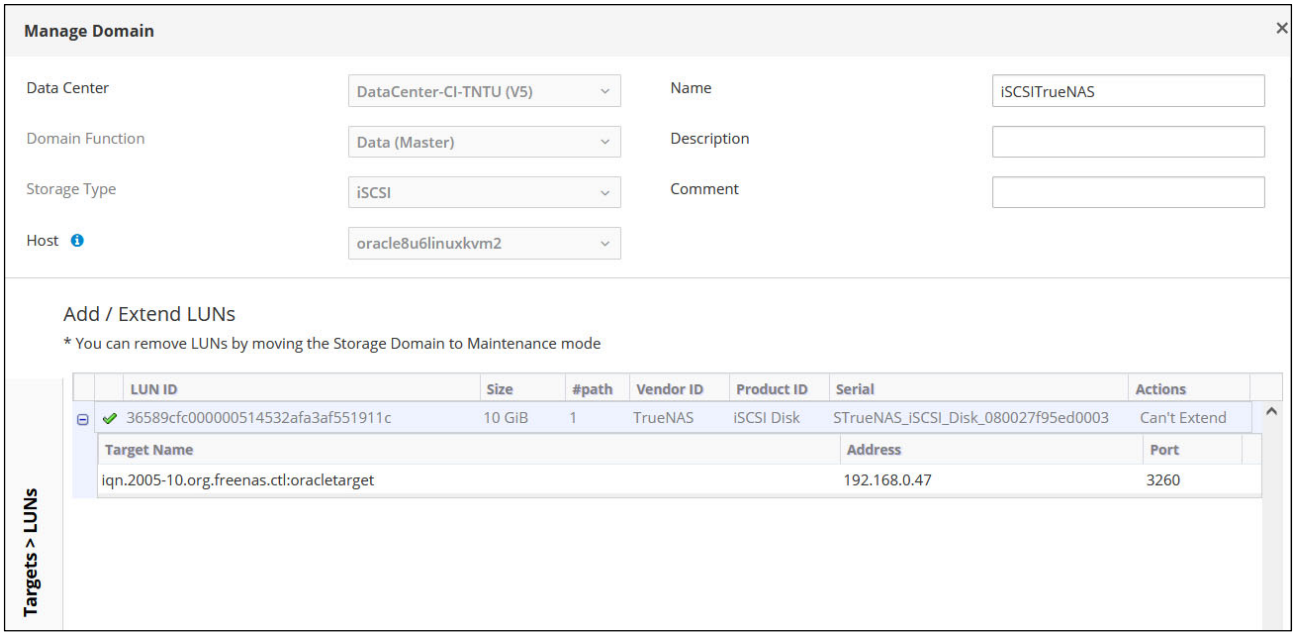


Рис. 3.13. Домен зберігання iSCSITrueNAS

3.2.7 Висока доступність і оптимізація. OLVM надає можливість налаштовувати кластер таким чином, щоб забезпечити оптимізацію роботи і високу доступність для віртуальних машин.

На рис.3.14 показано налаштування параметрів оптимізації для кластера Cluster-CI-TNTU. MoM є компонентом, який використовується для керування доступними ресурсами пам'яті в гіпервізорі. Одним із методів оптимізації роботи з пам'яттю є використання KSM. KSM - це механізм ядра Linux, який дозволяє злити однакові сторінки пам'яті (з однаковими вмістом) разом для зменшення дублювання і економії використання пам'яті. Коли в системі запускається KSM, він сканує процеси та їх пам'ять для знаходження ідентичних частин пам'яті. Знайдені однакові сторінки пам'яті об'єднуються, звільняючи зайві ресурси та зменшуючи використання оперативної пам'яті. MoM KSM, коли він визначає, що об'єднання сторінок пам'яті призведе до зменшення зайвого використання ресурсів пам'яті і покращить ефективність використання пам'яті гіпервізором. Це дозволяє системі більш ефективно використовувати доступну оперативну пам'ять, зменшуючи її загальне використання і звільнюючи ресурси для інших завдань.

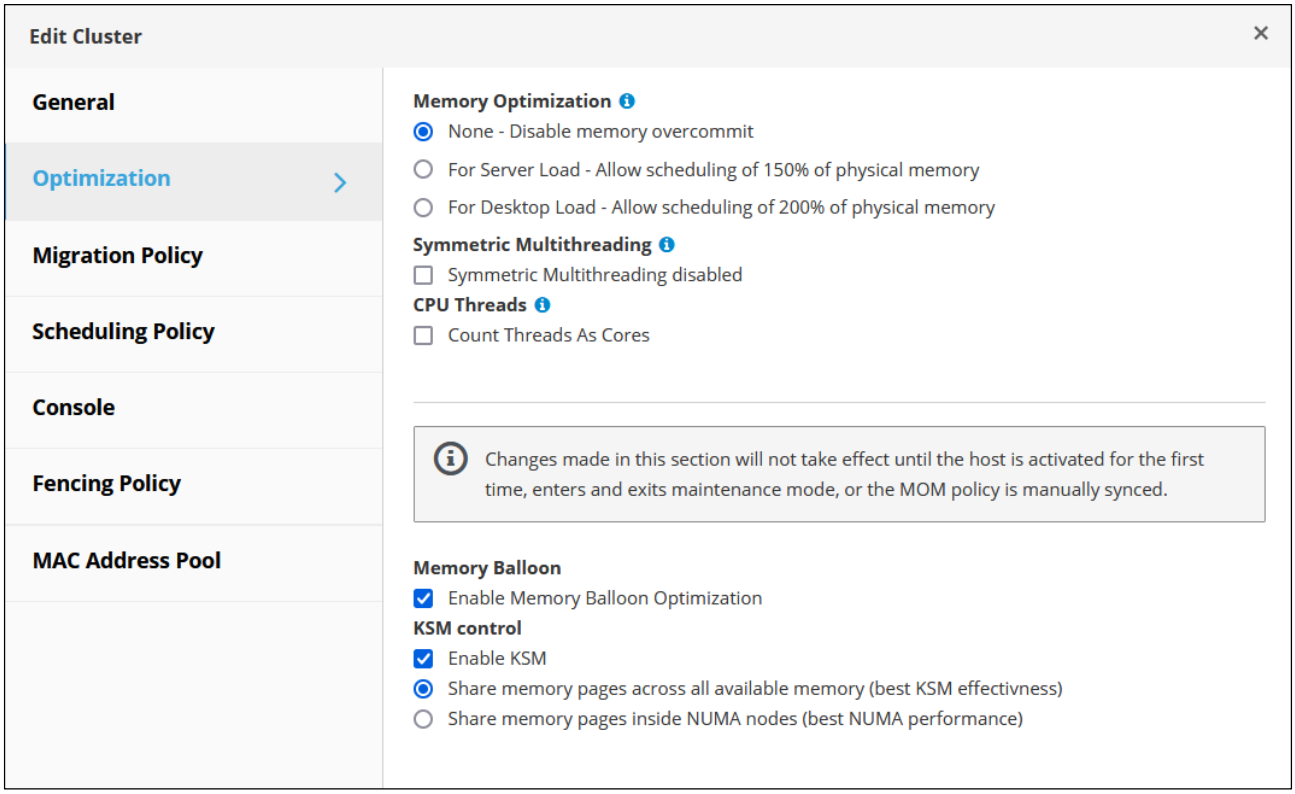


Рис. 3.14. Параметри оптимізації кластера Cluster-CI-TNTU

У віртуалізації та кластеризації механізм fencing гарантує, що жоден хост або віртуальна машина не продовжує роботу або не отримує доступ до спільних ресурсів після того, як він був визнаний як недоступний або відмовив. Це необхідно для забезпечення коректної роботи кластера та уникнення потенційних проблем. Fencing може включати в себе різні методи, такі як перезавантаження хоста, відключення від мережі або відключення доступу до спільних дисків. Його мета полягає у забезпеченні того, що інші частини системи вважають недоступний ресурс дійсно недоступним, дозволяючи роботі кластера продовжуватися без конфліктів чи непередбачуваних проблем. В разі виявлення недоступності хоста він застосовує визначені політики та заходи для забезпечення безперервної роботи системи та уникнення впливу недоступності на інші компоненти кластера. Ефективне управління ресурсами у віртуалізованому середовищі може бути досягнуте через планування, балансування навантаження та політику міграції віртуальних машин між різними хостами KVM у кластері (рис.3.15). Якщо один з хостів KVM перевантажений

або використовується надто інтенсивно, це може призвести до низької продуктивності віртуальних машин, які працюють на цьому хості. Для запобігання цьому використовують механізми автоматичного балансування навантаження, що включають політику міграції віртуальних машин. Якщо хост KVM надто навантажений, система автоматично може визначити цей стан і ініціювати процес міграції віртуальних машин на менш навантажений хост у кластері. Це дозволяє збалансувати навантаження та забезпечити оптимальне використання ресурсів між усіма хостами.

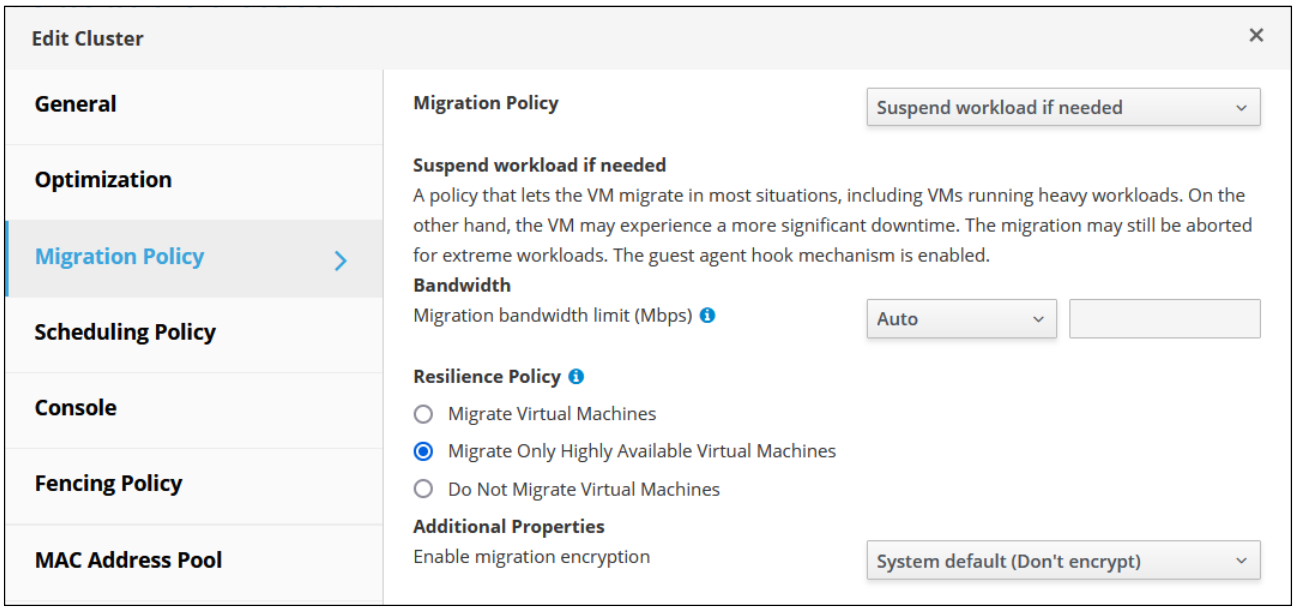


Рис. 3.15. Політика міграції в кластері Cluster-CI-TNTU

Якщо віртуальна машина налаштована для високої доступності у кластері і хост, на якому вона працює, виходить з ладу або стає недоступним, ця віртуальна машина автоматично переходить на інший доступний хост у кластері та перезапускається там, забезпечуючи продовження її роботи.

Однак, якщо віртуальна машина не налаштована для високої доступності і хост, на якому вона працює, вимикається вручну, ця віртуальна машина не переходить автоматично на інший доступний хост. В такому випадку вона залишається недоступною до тих пір, поки не буде вручну запущена на іншому хості або не будуть вжиті інші заходи для відновлення її роботи.

Віртуальні машини не переміщуються в режимі реального часу без належної конфігурації для такого переміщення, особливо без спільного сховища або налаштованого середовища для цього. Жива міграція віртуальних машин відбувається за потреби, коли виконуються певні умови, такі як політики енергозбереження, розподілу ресурсів або у випадку технічного обслуговування. У таких ситуаціях віртуальні машини можуть бути переміщені автоматично для забезпечення їх доступності та ефективного використання ресурсів.

На рис.3.16 показано налаштування параметрів високої доступності для віртуальної машини Mikrotik RouterOS CHR.

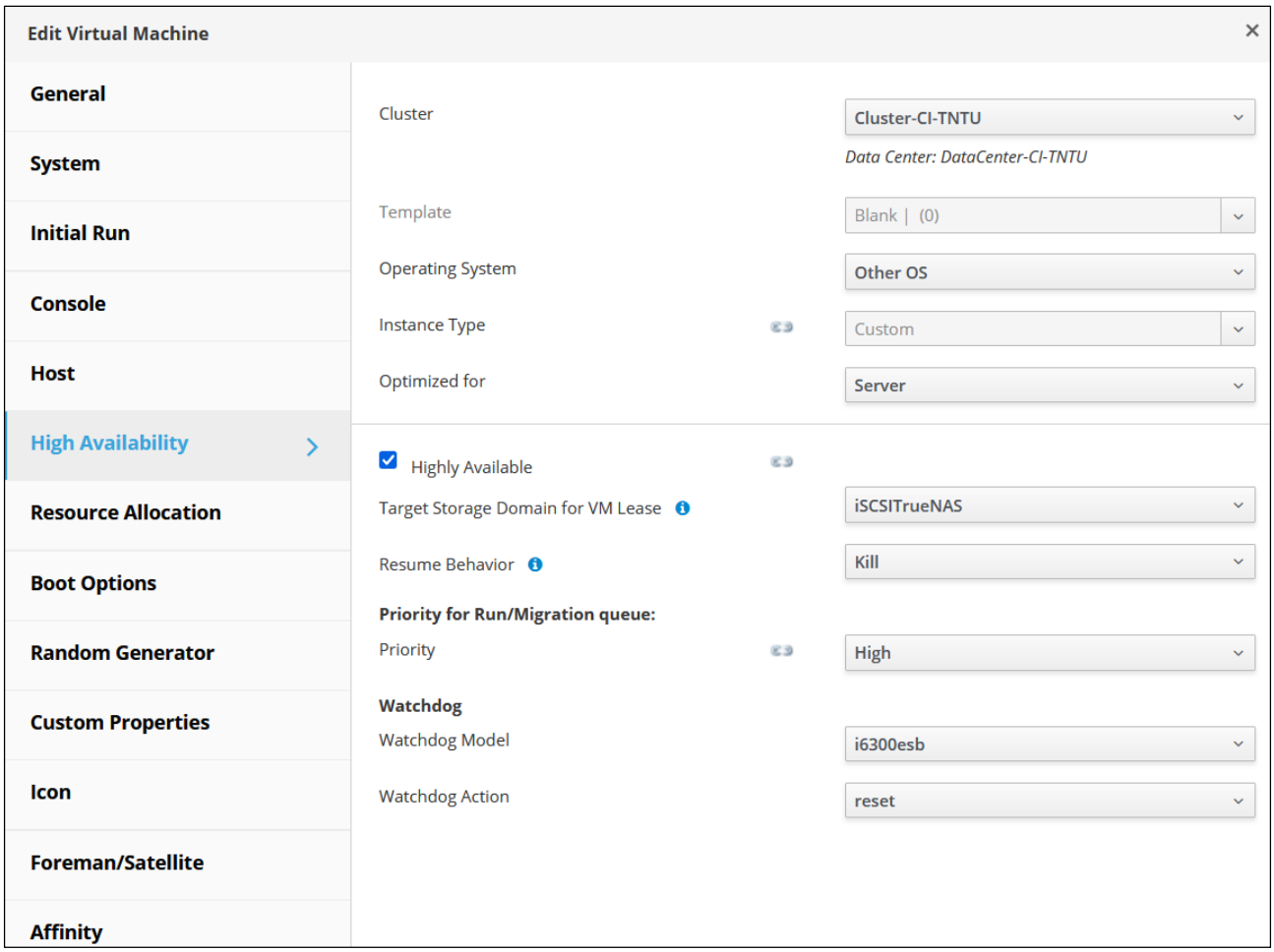


Рис. 3.16. Параметри високої доступності для віртуальної машини Mikrotik RouterOS CHR

Коли хост KVM переходить у режим обслуговування, віртуальні машини, які працювали на цьому хості, можуть бути автоматично переміщені на інші сервери в кластері. Це забезпечує уникнення простоїв віртуальних машин під час

запланованих періодів обслуговування, таких як оновлення програмного забезпечення, перезавантаження сервера або інші технічні процедури.

Коли віртуальна машина налаштована для високої доступності і її хост KVM виходить з ладу, система автоматично переносить цю віртуальну машину на інший доступний хост у кластері. Це забезпечує безперервну роботу віртуальних машин навіть у випадку відмови апаратного забезпечення.

Політики балансування навантаження, планування та відмовостійкості дозволяють критично важливим віртуальним машинам перезапускатися на іншому хості KVM у разі апаратного збою з трьома рівнями пріоритету.

Політики планування в OLVM дозволяють контролювати розподіл віртуальних машин між доступними хостами у кластері. За допомогою цих політик можна визначити, як система повинна реагувати на навантаження хостів, наприклад, встановити автоматичне балансування навантаження. Зазвичай відбувається автоматичне розподілення віртуальних машин між хостами для забезпечення рівномірного розподілу навантаження. Однак, у разі перевантаження конкретного хоста, віртуальні машини можуть бути переміщені на менш навантажений хост. Політики планування також можуть контролювати, як система реагує на перевантаження центрального процесора хоста. Наприклад, за замовчуванням, хост вважається перевантаженим, якщо його навантаження перевищує 80% протягом певного часу. Це значення можна змінювати через політики планування відповідно до вимог інфраструктури(рис.3.17).

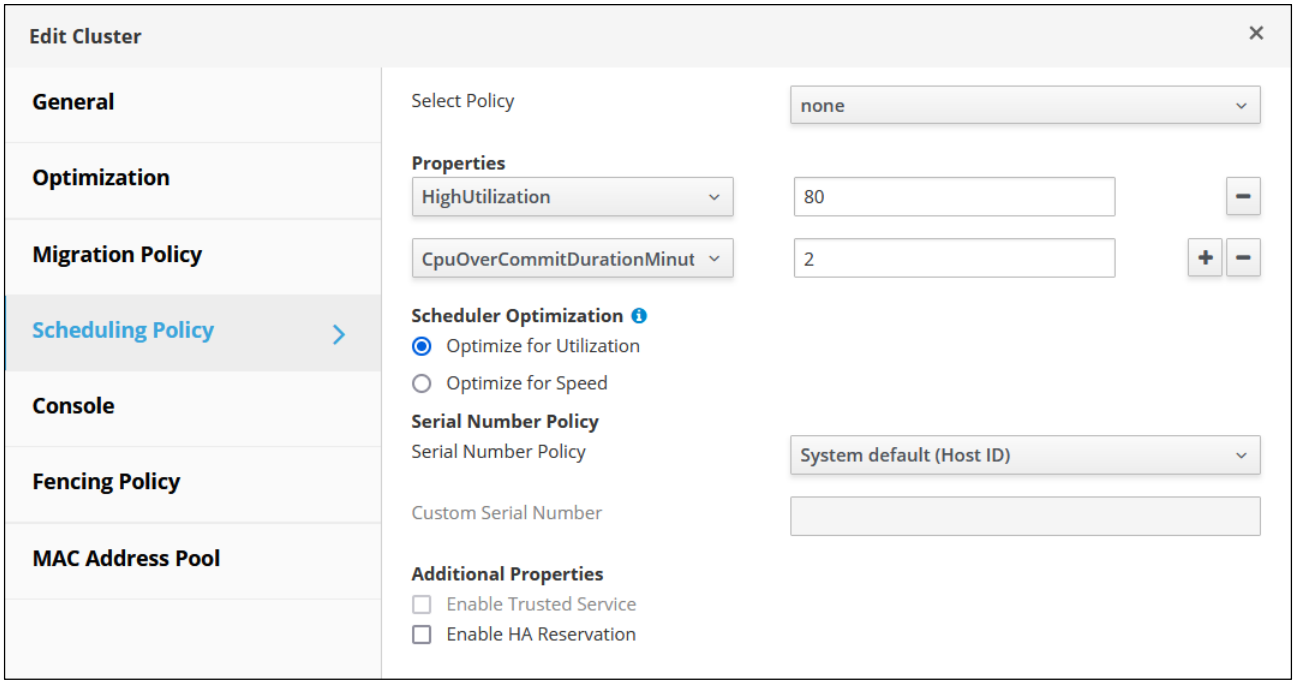


Рис. 3.17. Параметри політика планування в кластері Cluster-CI-TNTU

Політики міграції в OLVM дозволяють налаштовувати умови для переміщення віртуальних машин у реальному часі у випадку збою хоста KVM. Ці налаштування включають термін недоступності під час міграції віртуальної машини, обсяг пропускної здатності мережі, яка використовується для міграції, а також пріоритети віртуальних машин.

Політики стійкості в OLVM дозволяють встановлювати пріоритетність для віртуальних машин під час міграції. Це налаштування може бути сконфігуроване так, щоб усі віртуальні машини переміщувались, жодна з них не переміщувалась або переміщувалися лише високодоступні віртуальні машини (див.рис.3.15). Це допомагає уникнути перевантаження хостів та забезпечити стабільність системи під час міграції віртуальних машин.

3.3. Тестування кластера високої доступності

Тестування кластера високої доступності є важливим етапом перевірки надійності і стійкості системи. В додатку Б міститься блок-схема, яка описує алгоритм роботи кластера високої доступності. На етапі створення та

налаштування кластера було встановлено операційну систему Mikrotik RouterOS CHR та FreeBSD. Віртуальна машина Mikrotik налаштована для високої доступності у кластері Cluster-CI-TNTU. Операційну систему Mikrotik у віртуалізованому середовищі налаштовано як маршрутизатор з двома інтерфейсами. Один з інтерфейсів отримує по DHCP мережеві налаштування та IP-адресу з мережі 192.168.0.0/24. Інший інтерфейс є маршрутом за замовчуванням для віртуальної локальної мережі, та має статично встановлену IP-адресу 192.168.86.1/24 (рис.3.18).

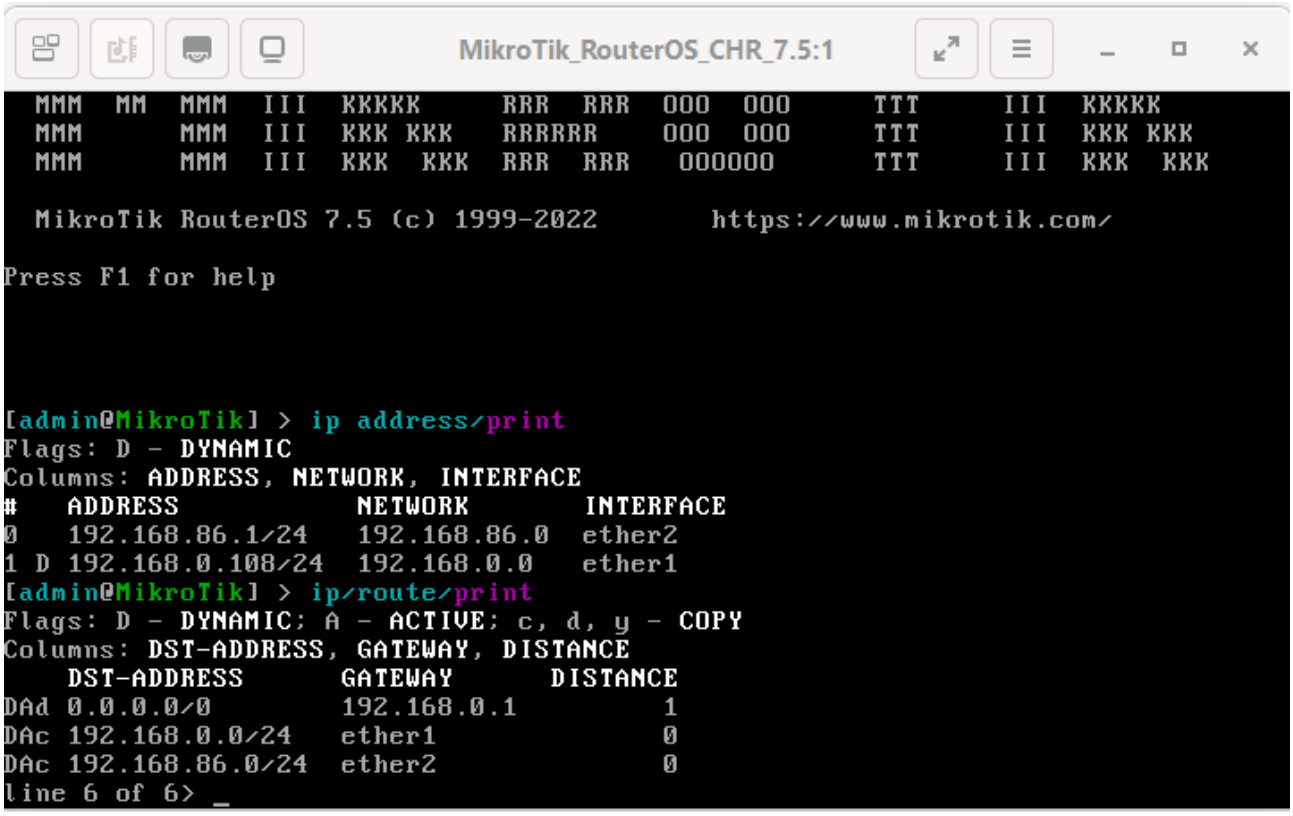


Рис. 3.18. Мережеві налаштування маршрутизатора Mikrotik RouterOS CHR

Віртуальна машина Mikrotik розміщена в мережевому сховищі даних на TrueNAS CORE (див.рис.3.10).

До віртуальної локальної мережі під'єднано сервер на базі операційної системи FreeBSD. З даного сервера буде проводитись перевірка доступності з'єднання з мережею Інтернет при різних сценаріях тестування кластера, які будуть впливати на роботу маршрутизатора Mikrotik.

Операційна система FreeBSD отримує мережеві налаштування по DHCP з маршрутизатора Mikrotik (рис.3.19).

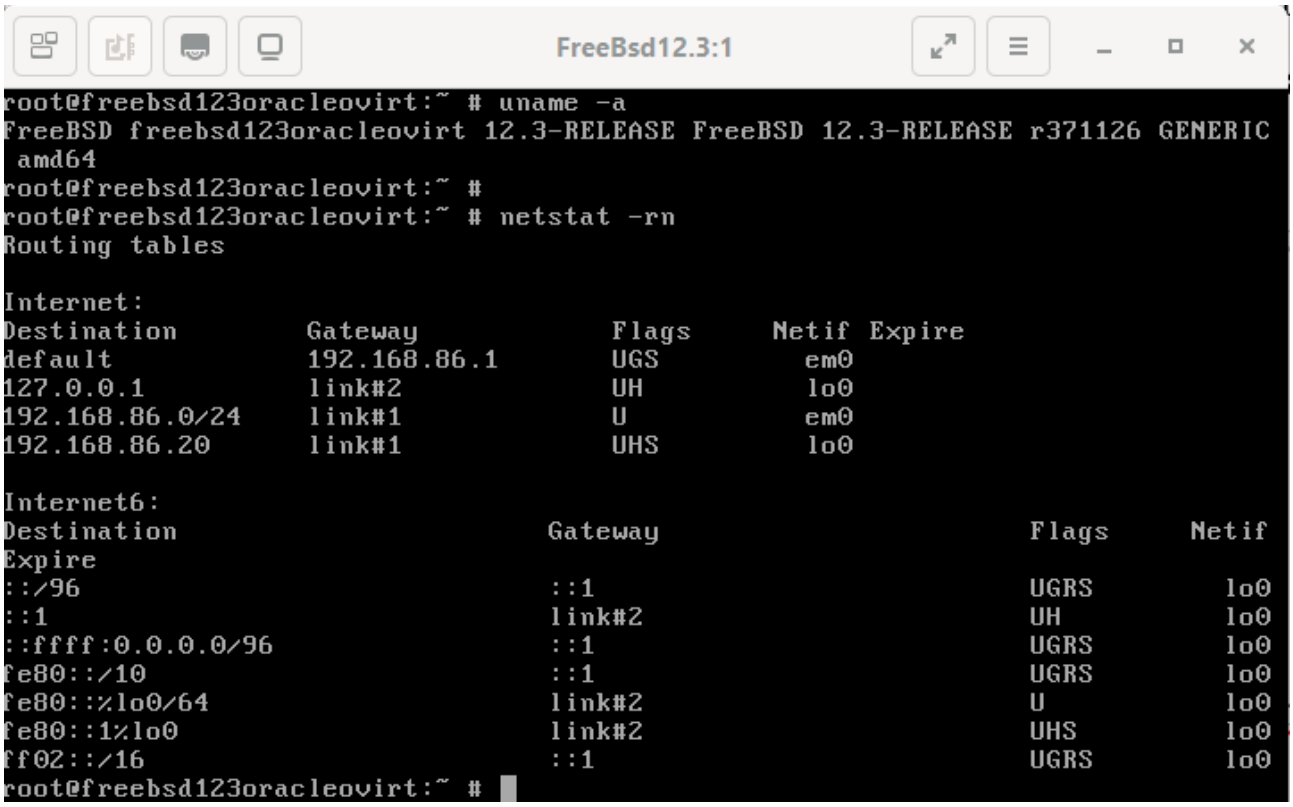


Рис. 3.19. Мережеві налаштування FreeBSD

На початковому етапі тестування маршрутизатор Mikrotik та FreeBSD працюють на KVM хості oracle8u6linuxkvm та oracle8u6linuxkvm2 відповідно (рис.3.20).

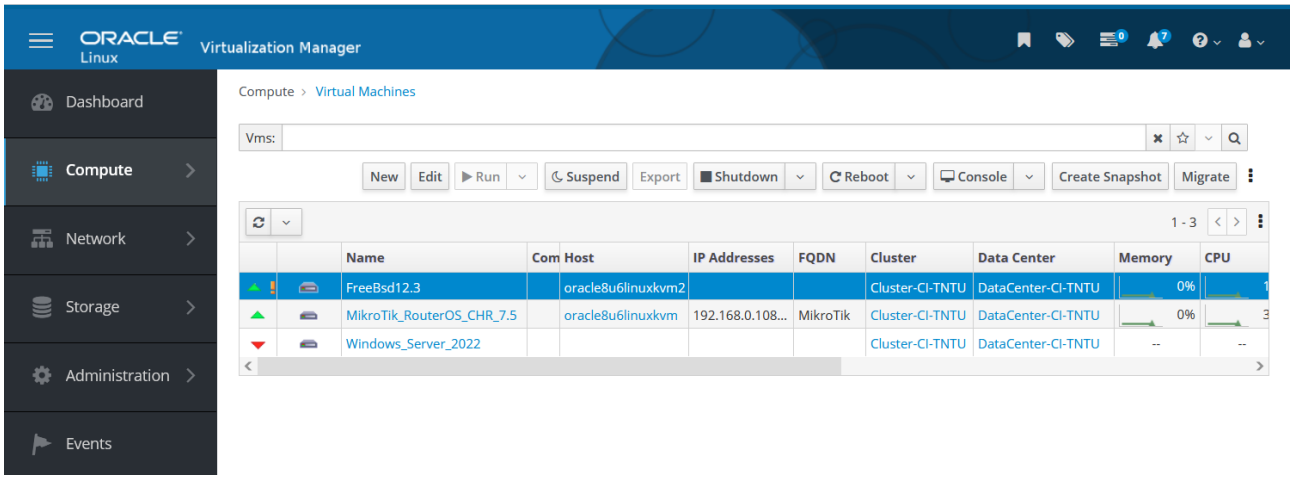


Рис. 3.20. Активні віртуальні машини в кластері

Доступ до мережі Інтернет з операційної системи FreeBSD працює коректно без втрат на каналі зв’язку (рис.3.21).

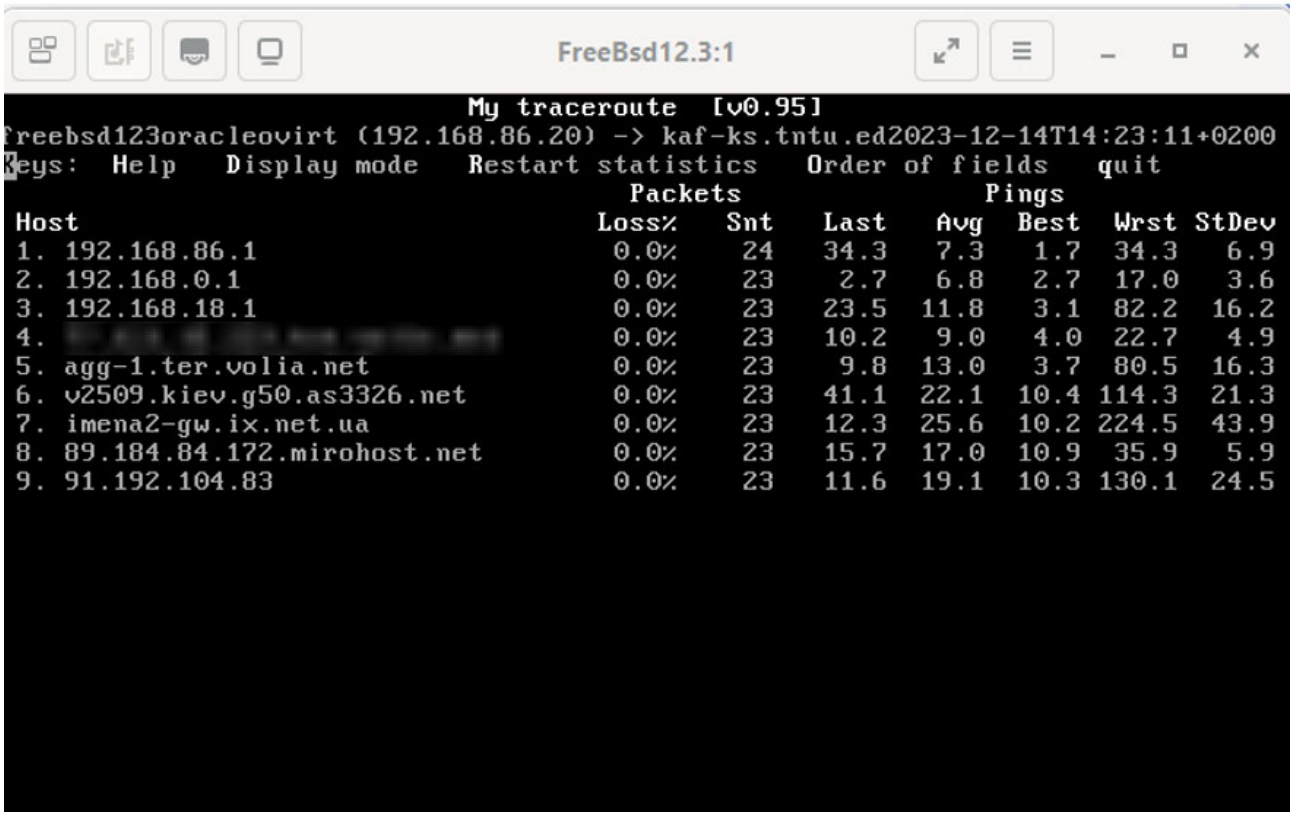


Рис. 3.21. Перевірка якості зв’язку до сайту kaf-ks.tntu.edu.ua

Віртуальні машини, що використовують одне сховище даних, можуть швидко переміщуватись між хостами у тому ж кластері без вимкнення. Жива міграція (live migration) дозволяє переміщувати активні віртуальні машини з одного фізичного сервера на інший без припинення роботи. Програми в межах віртуальної машини не зупиняються, продовжуючи працювати, поки машину переміщують на новий сервер. У цей час оперативна пам'ять копіюється з вихідного сервера на цільовий. Підключення до мережі та зберігання залишаються без змін.

Використання живої міграції дозволяє плавно переміщувати віртуальні машини для підтримки звичайних завдань обслуговування. Важливо

налаштувати середовище на підтримку цього процесу заздалегідь, до фактичного використання.

Для успішної живої міграції запущених віртуальних машин має бути виконано кілька умов:

- вихідний та цільовий хости повинні належати до одного кластера;

- обидва хости (вихідний та цільовий) повинні бути в активному стані;

- хости повинні мати доступ до тих самих віртуальних мереж і VLAN для забезпечення безперервного підключення;

- обидва хости повинні мати доступ до того ж домену зберігання даних (storage domain), де розташовані віртуальні машини;

- цільовий хост повинен мати достатню кількість потужності процесора та оперативної пам'яті, щоб задовольнити вимоги віртуальної машини після міграції.

На рис.3.22 показано процес живої міграції в ручному режимі з автоматичним вибором цільового хоста.

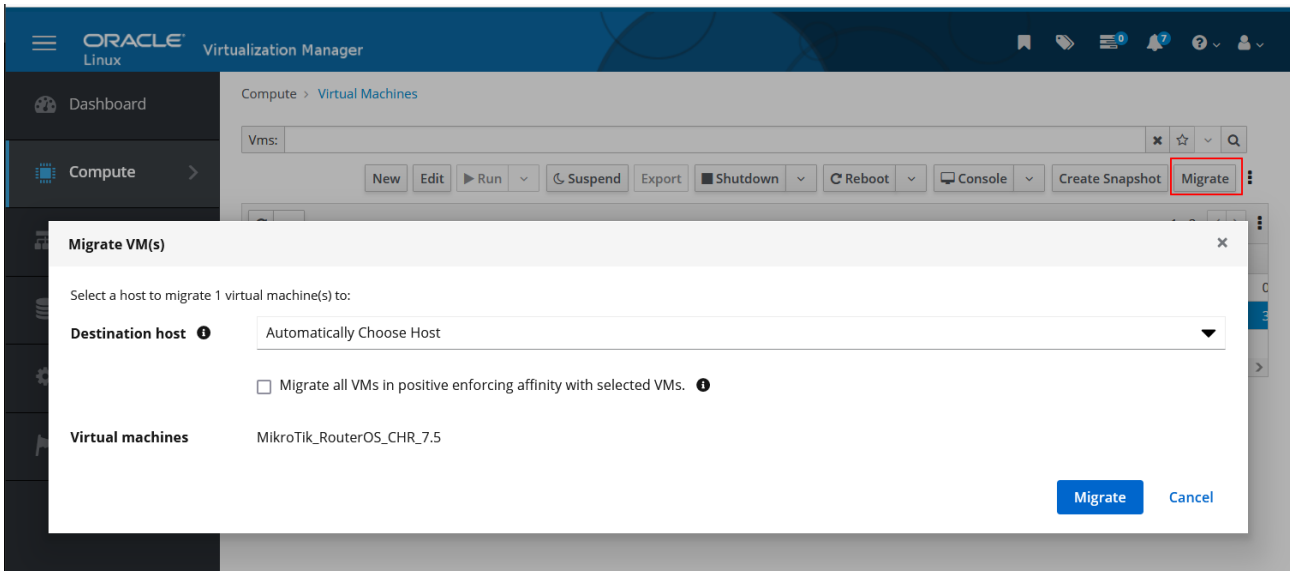


Рис. 3.22. Процес живої міграції в ручному режимі маршрутизатора Mikrotik

В файлі журналу OLVM можна побачити записи про результат виконання живої міграції в ручному режимі (рис.3.23).

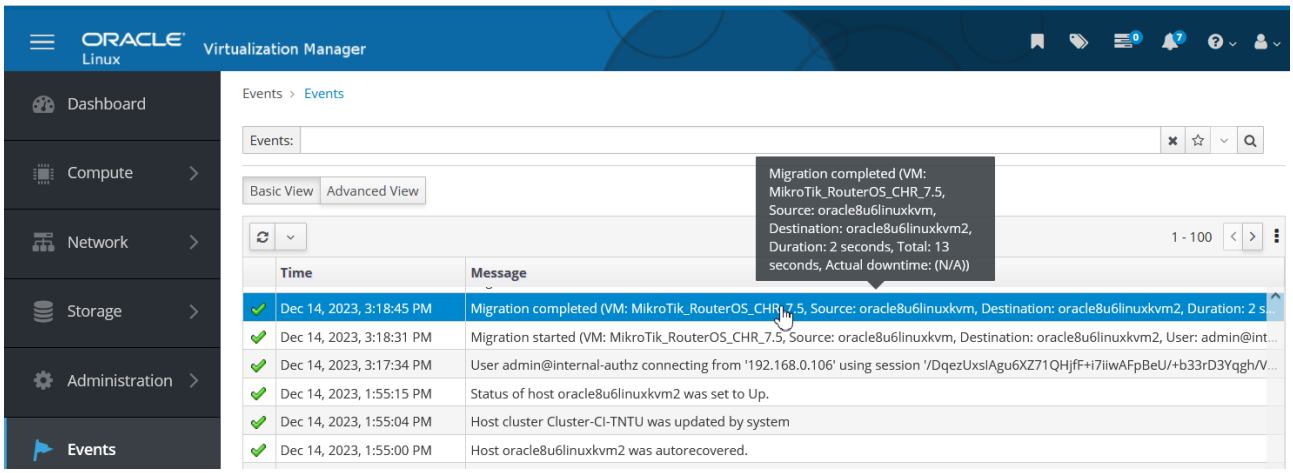


Рис.3.23. Записи файлу журналу про результат виконання живої міграції в ручному режимі

Процес міграції з вихідного хоста Oracle Linux KVM oracle8u6linuxkvm на цільовий хост oracle8u6linuxkvm2 завершився успішно. Міграція була завершена за 2 секунди, а весь процес, включаючи підготовку та міграцію, зайняв 13 секунд.

Це повідомлення підтверджує, що період простою (downtime) під час міграції був не виміряний (N/A), що свідчить про те, що процес міграції пройшов без простою, і віртуальна машина з назвою MikroTik_RouterOS_CHR_7.5 не мала зупинок або перебоїв під час перенесення між хостами.

OLVM автоматично спровокує живу міграцію віртуальних машин у двох ситуаціях. Перший – коли хост переходить у режим обслуговування (maintenance mode), тоді для всіх запущених віртуальних машин на цьому хості ініціюється оперативна міграція. Другий – для забезпечення рівноваги навантаження або для ефективного енергозбереження згідно з політикою планування, система спровокує живу міграцію віртуальних машин.

На рис.3.24 показано статус хоста в режимі обслуговування, що запустило процес автоматичної живої міграції з автоматичним вибором цільового хоста.

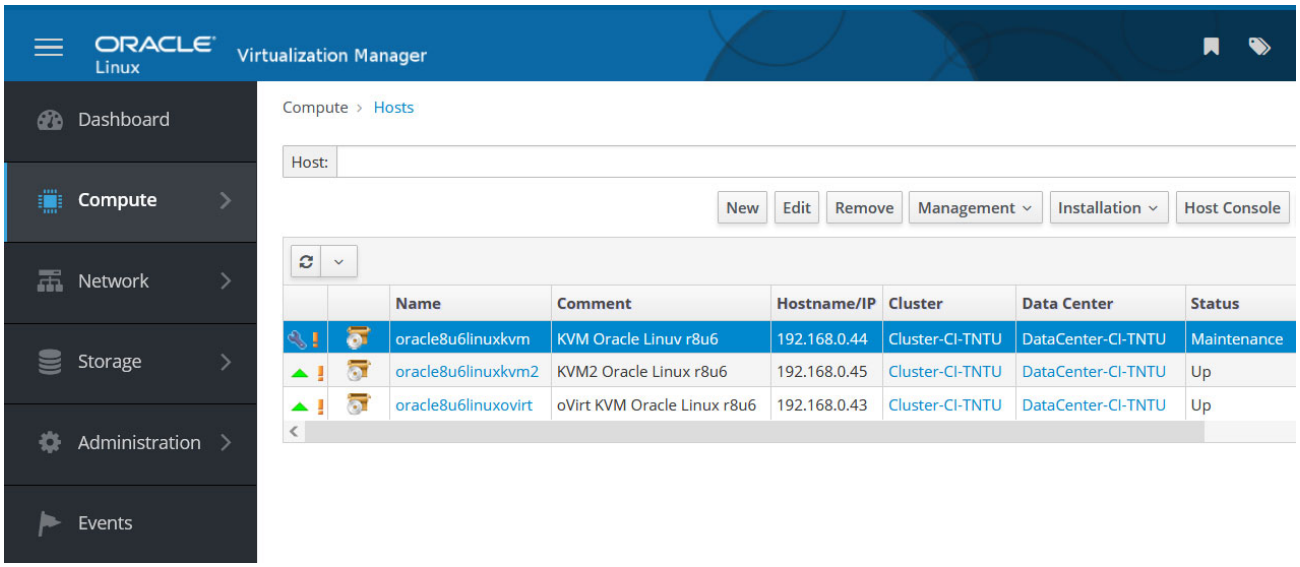


Рис.3.24. Статус хоста oracle8u6linuxkvm

В файлі журналу OLVM також можна побачити записи про результат виконання живої міграції в автоматичному режимі (рис.3.25).

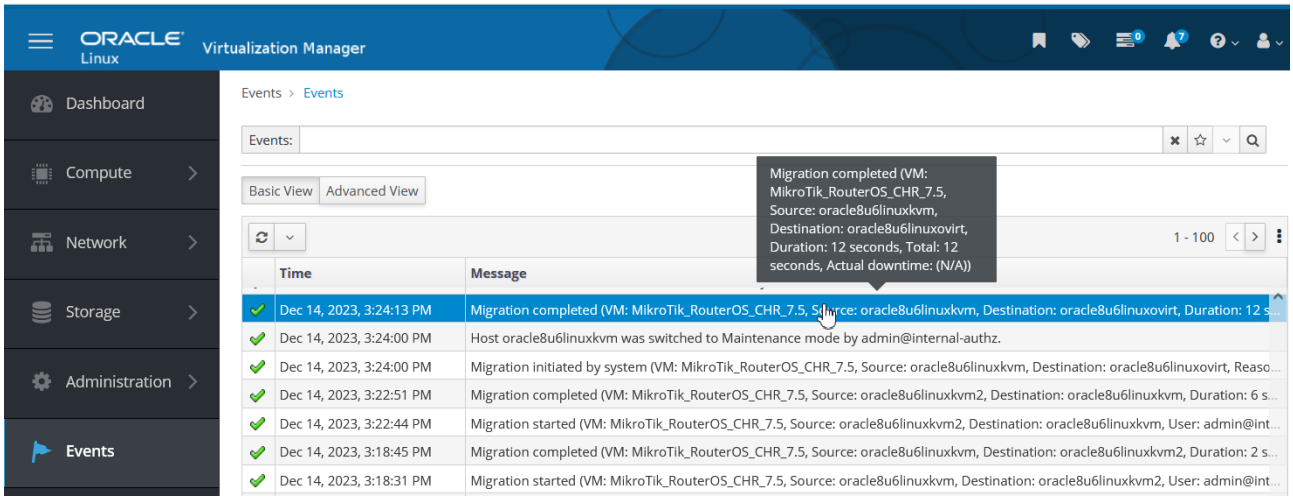


Рис.3.25. Записи файлу журналу про результат виконання оперативної живої міграції

Процес міграції віртуальної машини MikroTik_RouterOS_CHR_7.5 був ініційований системою через активацію режиму обслуговування на хості oracle8u6linuxkvm адміністратором admin@internal-authz.

Міграція успішно завершилася за 12 секунд, загальний час, затрачений на операцію, також склав 12 секунд. Операція не призвела до простою.

При відмові сервера (рис.3.26), на якому працюють віртуальні машини система автоматично переносить роботу на доступний сервер (рис.3.27).

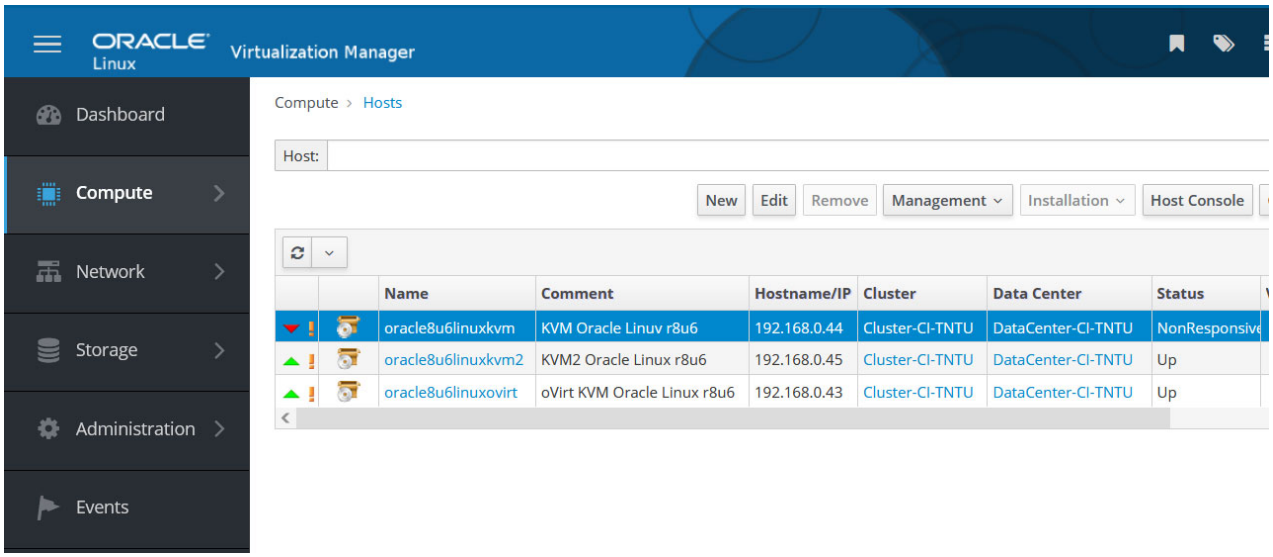


Рис.3.26. Відмова хоста oracle8u6linuxkvm

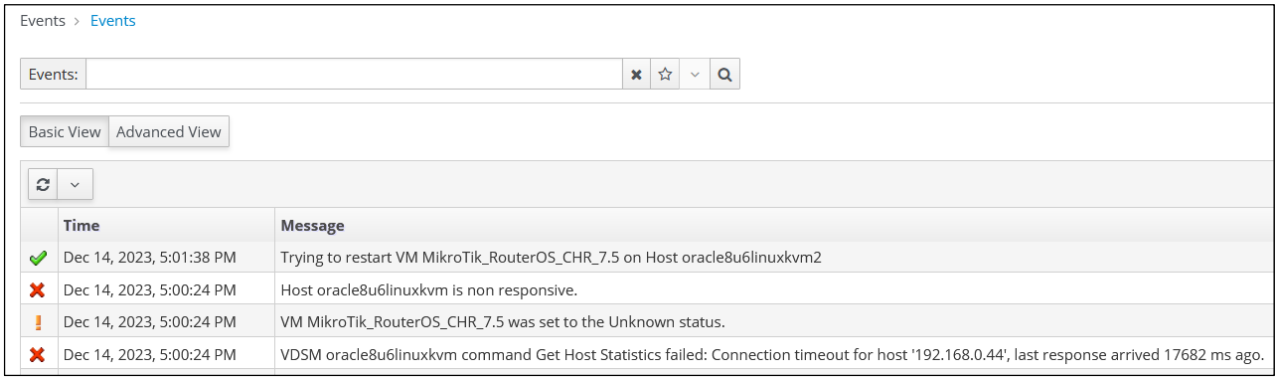


Рис.3.27. Записи файлу журналу про результат перезапуску віртуальної машини

Ця інформація є повідомленням про помилку в системі віртуалізації. Система спробувала отримати статистику з хоста з IP-адресою 192.168.0.44, проте отримала помилку connection timeout, оскільки не отримала відповіді вчасно. У зв'язку з недоступністю хоста oracle8u6linuxkvm віртуальна машина MikroTik_RouterOS_CHR_7.5 втратила свій статус і була переключена на статус unknown. Система автоматично відновила роботу цієї віртуальної машини, перезапустивши її на хості oracle8u6linuxkvm2.

Після перезапуску віртуальної машини Mikrotik можна переконатись що доступ до мережі Інтернет з операційної системи FreeBSD працює коректно без втрат на каналі зв’язку (рис.3.28).

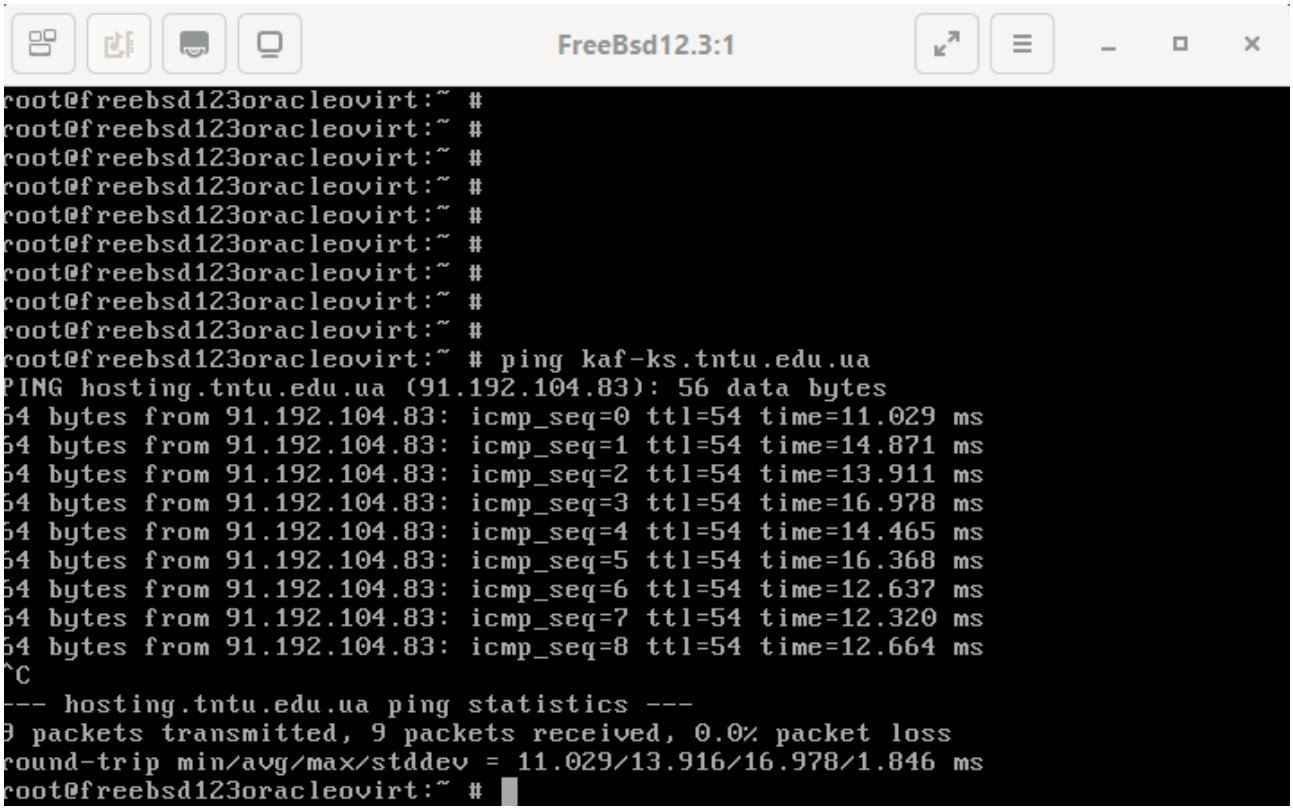


Рис. 3.28. Перевірка якості зв’язку після перезапуску віртуальної машини Mikrotik

Період простою віртуальної машини MikroTik_RouterOS_CHR_7.5 був відносно коротким, від моменту втрати зв'язку з першим хостом до перезапуску на другому хості пройшло менше двох хвилин.

Усі етапи перевірки кластера високої доступності були успішно завершені, це свідчить про те, що система може надійно керувати віртуальними ресурсами та забезпечувати стабільну роботу під час виникнення помилок або проблем з окремими хостами. Такий результат є важливим для забезпечення безперебійності роботи віртуальних машин та інфраструктури в цілому.

3.4. Висновки до розділу

Під час розробки кластера високої доступності була описана архітектура OLVM та створена схема взаємодії компонентів кластера, що базується на хостах Oracle Linux KVM та мережевому сховищі даних TrueNAS CORE. Побудовано кластер високої доступності та проведено процес тестування, спрямований на перевірку надійності та стійкості системи. В результаті підтвердилося, що система здатна ефективно керувати віртуальними ресурсами та забезпечувати стабільну роботу навіть під час помилок або проблем з окремими хостами.

РОЗДІЛ 4
ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Охорона праці

У кваліфікаційній роботі магістра проводиться дослідження методів і засобів створення систем високої доступності на основі кластерів . Оскільки, такі роботи передбачають використання комп’ютерної техніки, зокрема ПК та серверного обладнання, то обов’язком виконавця такого процесу є дотримання вимог охорони праці і техніки безпеки.

Роботодавець зобов'язаний згідно Закону України «Про охорону праці» стаття 13 «Управління охороною праці та обов’язки роботодавця» створити на робочому місці в кожному структурному підрозділі умови праці відповідно до нормативно-правових актів, а також забезпечити додержання вимог законодавства щодо прав працівників у галузі охорони праці [16].

Із цією метою роботодавець забезпечує функціонування системи управління охороною праці, а саме:

- створює відповідні служби і призначає посадових осіб, які забезпечують вирішення конкретних питань охорони праці, затверджує інструкції про їхні обов'язки, права та відповідальність за виконання покладених на них функцій, а також контролює їх додержання;

- розробляє за участю сторін колективного договору і реалізує комплексні заходи для досягнення встановлених нормативів та підвищення існуючого рівня охорони праці;

- забезпечує виконання необхідних профілактичних заходів відповідно до обставин, що змінюються;

- впроваджує прогресивні технології, досягнення науки і техніки, засоби механізації та автоматизації виробництва, вимоги ергономіки, позитивний досвід з охорони праці тощо;

- забезпечує належне утримання будівель та споруд, виробничого обладнання та устаткування, моніторинг за їх технічним станом;

- забезпечує усунення причин, що призводять до нещасних випадків, професійних захворювань, та здійснення профілактичних заходів, визначених комісіями за підсумками розслідування цих причин;

- організовує проведення аудиту охорони праці, лабораторних досліджень умов праці, оцінку технічного стану виробничого обладнання та устаткування, атестацій робочих місць на відповідність нормативно-правовим актам з охорони праці в порядку і строки, що визначаються законодавством, та за їх підсумками вживає заходів з усунення небезпечних і шкідливих для здоров'я виробничих факторів;

- розробляє і затверджує положення, інструкції, інші акти з охорони праці, що діють у межах підприємства та встановлюють правила виконання робіт і поведінки працівників на території підприємства, у виробничих приміщеннях, на будівельних майданчиках, робочих місцях відповідно до нормативно-правових актів з охорони праці, забезпечує безоплатно працівників нормативно-правовими актами підприємства з охорони праці;

- здійснює контроль за дотриманням працівником технологічних процесів, правил поведінки з машинами, механізмами, устаткуванням та іншими засобами виробництва, використанням засобів колективного та індивідуального захисту, виконанням робіт відповідно до вимог з охорони праці;

- організовує пропаганду безпечних методів праці та співробітництво з працівниками у галузі охорони праці.

Роботодавець несе безпосередню відповідальність за порушення нормативно-правових актів з охорони праці.

Для забезпечення оптимальних умов праці працівників при розробці, встановлення та налаштуванні систем високої доступності на основі кластерів необхідно передбачити відповідність мікроклімату у приміщеннях згідно вимог ДСН 3.3.6.042-99. Категорія робіт при виконанні даного виду завдань належить до легкої – Іб. Для того щоб визначити, чи відповідає повітря приміщення

встановленим нормам, необхідно кількісно оцінити кожний з його параметрів. Оптимальні показники мікроклімату, які необхідно забезпечити у приміщеннях, де експлуатуються ПК у теплу пору року повинні становити: температура – +22 оС - +24 оС, відносна вологість – 40-60%, швидкість руху повітря 0,1 м/с.

Окрім, забезпечення оптимальних показників мікроклімату, необхідно передбачити ще й оптимальні показники шуму та вібрації на робочих місцях.

Граничні величини шуму на робочих місцях регламентуються ДСН 3.3.6.037 – 99 „Державні санітарні норми виробничого шуму, ультразвуку та інфразвуку". В ньому закладено принцип встановлення певних параметрів шуму, виходячи з класифікації приміщень та їх використання для трудової діяльності. Окрім цього, на робочих місцях працівників необхідно забезпечити дотримання вимог НПАОП 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями».

Основними вимогами, визначеними у цьому нормативному документі є:

- площу та об'єм для одного робочого місця оператора визначають згідно з вимогами ДСанПіН 3.3.2-007-98. Площа має бути не менше 6,0 кв.м, об'єм – не менше 20,0 куб.м.;

- заземлені конструкції, що знаходяться в приміщеннях, де розміщені робочі місця операторів (батареї опалення, водопровідні труби, кабелі із заземленим відкритим екраном), мають бути надійно захищені діелектричними щитками або сітками з метою недопущення потрапляння працівника під напругу;

- приміщення, де розміщені робочі місця операторів, крім приміщень, у яких розміщені робочі місця операторів великих ЕОМ загального призначення (серверні), повинні бути оснащені системою автоматичної пожежної сигналізації.

Дотримання даних вимог є необхідними при створенні системи високої доступності на основі кластерів з використанням серверного обладнання та ПК.

4.2. Безпека в надзвичайних ситуаціях

4.2.1. Державна система моніторингу довкілля, як складова частина національної інформаційної інфраструктури, сумісної з аналогічними системами інших країн.

Законодавство України, а саме Статті 20 та 22 Закону України «Про охорону навколишнього природного середовища» [17], передбачають створення державної системи моніторингу довкілля та проведення спостережень за станом навколишнього природного середовища і рівнем його забруднення. Основні принципи функціонування державної системи моніторингу довкілля визначені у постанові Кабінету Міністрів України від 30.03.1998 №391 «Про затвердження Положення про державну систему моніторингу довкілля» [18]. Згідно з цим положенням, «Державна система моніторингу довкілля (ДСМД) – це система спостережень, збирання, оброблення, передавання, збереження та аналізу інформації про стан довкілля, прогнозування його змін і розроблення науково-обґрунтованих рекомендацій для прийняття рішень про запобігання негативним змінам стану довкілля та дотримання вимог екологічної безпеки».

Також визначається, що система моніторингу є відкритою інформаційною системою, складовою частиною національної інфраструктури, сумісної з аналогічними системами інших країн.

Пріоритетами такої системи є збереження екосистем, відвернення кризових змін в екологічному стані довкілля та запобігання відповідним надзвичайним ситуаціям. На даний час, у ДСМД функції і задачі спостережень та інформаційного забезпечення виконують такі суб`єкти системи моніторингу [18]:

- Міністерство захисту довкілля та природних ресурсів України;
- Міністерство аграрної політики та продовольства України;
- Міністерство розвитку громад і територій України;
- Державна служба з надзвичайних ситуацій;
- Державна служба геології та надр України;

- Державне агентство України з управління зоною відчуження;
- Державне агентство лісових ресурсів України;
- Державне агентство водних ресурсів України;
- Державна служба України з питань геодезії, картографії та кадастру;
- Державне космічне агентство України.

Окрім цього, виконання таких функцій покладено на інші центральні органи виконавчої влади, які є суб'єктами державної системи моніторингу довкілля, а також на підприємства, установи та організації, діяльність яких призводить або може призвести до погіршення стану довкілля. В загальному, система моніторингу спрямована на декілька основних цілей, серед яких підвищення рівня вивчення і знань про екологічний стан довкілля, зростання якості інформаційного обслуговування користувачів на всіх рівнях, покращення якості обґрунтування природоохоронних заходів та ефективності їх здійснення, а також сприяння розвитку міжнародного співробітництва у галузі охорони довкілля, раціонального використання природних ресурсів та екологічної безпеки [18].

Для досягнення поставлених цілей, положення про ДСМД визначає такі завдання: систематичні спостереження за станом довкілля, аналіз стану навколишнього середовища, прогнозування змін довкілля, інформаційна підтримка прийняття рішень, забезпечення органів державної та місцевої влади, населення та партнерів інформацією про актуальний стан довкілля [18]. Також згідно з положенням складовими частинами державного моніторингу навколишнього середовища України є моніторинг атмосферного повітря, води, земель, біологічного різноманіття, лісів, відходів, геологічного середовища, фізичних факторів впливу.

Нормативними актами, що регламентують моніторинг таких об’єктів є відповідні постанови Кабінету Міністрів України щодо порядків здійснення моніторингу повітря, вод, земель та ґрунтів. Система моніторингу ґрунтується на використанні існуючих організаційних структур суб'єктів моніторингу і функціонує на основі єдиного нормативного, організаційного, методологічного і метрологічного забезпечення, об'єднання складових частин та уніфікованих

компонентів цієї системи [19]. Зазначимо, що функціонування ДСМД здійснюється на трьох рівнях, які розподіляються за територіальним принципом, а саме: загальнодержавний, регіональний та локальний рівні.

Відповідальні суб'єкти здійснюють моніторинг різного роду об'єктів, серед яких [18]:

- ґрунти на природоохоронних територіях, а також ґрунти сільськогосподарського та лісового фондів;

- види рослинного і тваринного світу, що перебувають під загрозою зникнення чи під особливою охороною;

- вміст радіонуклідів в атмосферному повітрі, водах та ґрунтах;

- наявність та серйозність повеней, паводків, снігових лавин, селів;

- об'єкти зберігання та захоронення радіоактивних відходів;

- сільськогосподарські рослини, тварини і продуктів з них, мисливська фауна та лісова рослинність;

- якість вод водогосподарських систем міжгалузевого та сільськогосподарського водопостачання;

- зрошувані та осушувані землі у сенсі глибини залягання та мінералізації ґрунтових вод, ступені засоленості та солонцюватості ґрунтів;

- ґрунти і ландшафти щодо проявів ерозійних та інших екзогенних процесів та просторового забруднення земель об'єктами промислового і сільськогосподарського виробництва;

- берегові лінії річок, морів, озер, водосховищ, лиманів, заток, гідротехнічних споруд;

- стічні води міської каналізаційної мережі та очисні споруди, джерела скидання таких вод;

- зелені насадження у містах і селищах міського типу.

Якщо розглянути моніторинг повітря, то Державною гідрометеорологічною службою здійснюється спостереження за забрудненням атмосферного повітря у містах України. Державна екологічна інспекція здійснює відбір проб на джерелах викидів, а санітарно-епідеміологічна служба координує моніторинг

якості атмосферного повітря у житлових зонах. Контроль якості повітря також включає аналіз опадів та снігового покрову. Програма обов`язкового моніторингу якості атмосферного повітря охоплює сім забруднюючих речовин: пил, двоокис азоту, двоокис сірки, оксид вуглецю, формальдегід, свинець та бензапірен. Спостереження за водами суші на 151 об'єкті проводить Державна гідрометеорологічна служба, що включає в себе оцінку хімічного складу вод, біогенних параметрів, наявність зважених часток та органічних речовин, основних забруднюючих речовин, важких металів та пестицидів. Контроль за водами суші також здійснюють Державна екологічна інспекція, Державний комітет по водному господарству, Санітарно-епідеміологічна служба та Державна геологічна служба. Дослідження включають моніторинг річок, водосховищ, каналів тощо, контроль хімічних, радіаційних та фізичних показників, а також придатності води до споживання. За схожими параметрами відбувається й моніторинг прибережних вод. До моніторингу ґрунтів входить вимірювання забруднення ґрунтів пестицидами, агровідходами, токсинами та важкими металами на сільськогосподарських землях та промислових майданчиках. Також досліджується забруднення ґрунту у місцях захоронення відходів. Контроль здійснюється державною гідрометеорологічною службою, Міністерством захисту довкілля та Міністерством аграрної політики. Моніторинг радіаційного випромінювання включає в себе спостереження за радіоактивним забрудненням атмосфери, поверхневих вод та ґрунтів поблизу атомних електростанцій та у зоні відчуження [19].

Згідно з положенням [18] суб'єкти системи моніторингу інформаційно підтримують рішення в галузі охорони довкілля, безкоштовно обмінюються результатами спостережень на об'єктах та колективно використовують інформаційні ресурси, надаючи всім зацікавленим сторонам відповідні дані.

4.2.2. Оцінка стійкості роботи промислового підприємства до дії світлового випромінювання ядерного вибуху.

Основну небезпеку для наземних об'єктів, зокрема і телекомунікаційних, становлять ударна хвиля, світлове (теплове) випромінювання, вторинні вражаючі фактори і радіоактивне зараження місцевості. Проте іноді доводиться враховувати і вплив проникаючої радіації та електромагнітного імпульсу.

Критеріями оцінки фізичної стійкості об'єкта прийняті: при впливі ударної хвилі - надлишкові тиски, при яких елементи виробничого комплексу не руйнуються або одержують такі ушкодження чи руйнування (слабкі і середні), при яких вони можуть бути відновлені в короткі терміни; при впливі світлового випромінювання - максимальні значення світлових імпульсів, при яких не відбувається загоряння матеріалів, сировини, устаткування, будинків і споруд; при впливі вторинних факторів - надлишкові тиски, при яких руйнування і пошкодження не призводять до аварій, пожеж, вибухів, затоплень, небезпечного зараження місцевості й атмосфери, тобто не призводять до ураження людей і виходу з ладу засобів виробництва [19].

Оцінка стійкості об'єкта включає визначення: видів вражаючих факторів, вплив яких можливий на об'єкт, та їх параметрів; впливу ударної хвилі на елементи об'єкта; можливості виникнення пожеж; впливу вторинних вражаючих факторів.

Світлове випромінювання - це електромагнітне випромінювання, основним джерелом якого є світна область вибуху (вогненна куля), що складається з розпечених продуктів вибуху і повітря. Температура в ній сягає від 6 тисяч градусів за С. Тривалість світіння залежить від потужності ядерного заряду: при вибуху малого калібру — 1-2 сек., середнього — 2-4 сек, крупного та надкрупного — 10 і більше сек.

На світлове випромінювання припадає приблизно 30 % всієї енергії ядерного вибуху. Воно складається з ультрафіолетових, інфрачервоних і видимих променів. Основна кількість енергії світлового випромінювання (85%) виділяється в перші секунди з моменту вибуху.

Кількість енергії світлового випромінювання, яке падає на 1 см 2 поверхні, перпендикулярної напрямку його поширення, за весь час світіння, називається світловим імпульсом. Його величина вимірюється в джоулях на квадратний метр (Дж/м2).

Уражуюча дія світлового випромінювання вимірюється, головним чином, величиною світлового імпульсу і часом дії. Чим більша величина світлового імпульсу, що випромінюється за менший час, тим сильніший уражуючий ефект, який пропорційний поглинутій кількості енергії. Остання перетворюється в тепло і здатна викликати опіки та приводити до спалахування різних предметів.

Ураження людини можливе, як в результаті безпосередньої дії світлового випромінювання на шкірні покриви - світлові (первинні) опіки, так і в результаті спалахування одягу і навколишніх предметів - опосередковані (вторинні) опіки.

Можливість виникнення пожеж встановлюють за займистістю матеріалів від світлового імпульсу ядерного вибуху, руйнування печей, газопроводів, пошкодження електромережі, які можуть виникнути при аваріях, землетрусах, бурях та ін.

При оцінці стійкості об'єкта проти світлового випромінювання ядерного вибуху необхідно визначити максимальне значення світлового імпульсу яке може бути на об'єкті. Світловий імпульс можна розрахувати за температурою загорання або нагрівання матеріалів і виробів.

Для оцінки стійкості об'єкта проти світлового випромінювання необхідні такі вихідні дані: характеристика будівель і споруд, характер виробництва, які горючі матеріали застосовуються у виробництві; вид готової продукції та місце її зберігання.

Оцінку стійкості електроенергетичного об'єкта до світлового випромінювання ядерного вибуху доцільно проводити у такій послідовності:

- визначити мінімальну відстань до можливого центру вибуху Rx, км;
- визначити максимальне значення світлового імпульсу I св.max, кДж/м2;
- визначити ступінь вогнестійкості будинку цеху;

- виявити в конструкціях будівлі елементи, виготовлені із горючих матеріалів та визначити їх характеристики (наприклад: для дверей та віконних рам – дерев'яних, пофарбованих в темний колір I св.max =250, кДж/м2);

- визначити границю стійкості об'єкту до світлового випромінювання за мінімальним світловим імпульсом, що викликає спалахування в будівлі I св.lim. кДж/м2;

- здійснити порівняння та дати оцінку стійкості об'єкту: при I св.lim < I св.max об'єкт не стійкий до світлового випромінювання, при I св.lim > I св.max - стійкий;

- визначити ступінь руйнування будівлі від ударної хвилі при максимальному очікуваному надлишковому тиску;

- визначити зону пожеж, в якій може опинитися об'єкт.

На основі отриманих даних робиться відповідний висновок щодо стійкості електроенергетичного об'єкта до світлового випромінювання ядерного вибуху:

- чи викликає складну пожежну ситуацію на об'єкті при ядерному вибуху заданої потужності очікуваний максимальний світловий імпульс (кДж/м2) та надлишковий тиск ударної хвилі (кПа);

- чи опиниться об'єкт в зоні суцільних пожеж;

- чи об'єкт стійкий до світлового випромінювання за границею стійкості;

- що із обладнання, конструктивних елементів будівлі складає пожежну небезпеку для об'єкту;

- робитися висновок про доцільність підвищення границі стійкості об'єкту.

Підвищити границю стійкості можливо шляхом виконання наступних заходів: замінити крівлю об'єкту на азбестоцементові; замінити дерев'яні віконні рами на металеві; набити на двері сталлю по азбестовій прокладці; провести на об'єкті профілактичні протипожежні заходи.

4.3. Висновки до розділу

У цьому розділі було детально розглянуто дотримання вимог охорони праці і техніки безпеки при створенні системи високої доступності на основі кластерів з використанням серверного обладнання та ПК. У частині безпеки в надзвичайних ситуаціях було розглянуто питання державної служби моніторингу довкілля, а саме її основних цілей та задач, суб’єктів та об’єктів моніторингу та описано механізм оцінка стійкості роботи промислового підприємства до дії світлового випромінювання від ядерного вибуху.

ВИСНОВКИ

Під час проведення дослідження для магістерської кваліфікаційної роботи були вивчені та оцінені методи та інструменти віртуалізації, спрямовані на створення високодоступних комп'ютерних систем. У підсумку виконаних теоретичних та практичних досліджень були отримані такі результати:

- проведено огляд та порівняння сучасних методів створення кластерів на основі платформ віртуалізації VMware ESXi, Microsoft Hyper-V, Citrix Hypervisor та Oracle Linux KVM;

- досліджено можливості платформ віртуалізації для забезпечення високої доступності системи у випадку відмови одного чи кількох серверів;

- оцінено можливість використання гіпервізора з відкритим кодом KVM для створення кластерів високої доступності;

- розглянуто принципи та особливості роботи гіпервізора KVM;

- розглянуто взаємодію KVM та QEMU при використанні апаратної віртуалізації;

- описана архітектура Oracle Linux Virtualization Manager;

- реалізовано схему взаємодії компонентів кластера, що базується на хостах Oracle Linux KVM та мережевому сховищі даних TrueNAS CORE;

- побудовано кластер високої доступності та проведено процес тестування, спрямований на перевірку надійності та стійкості системи;

- підтверджено що система здатна ефективно керувати віртуальними ресурсами та забезпечувати стабільну роботу навіть під час помилок або проблем з окремими хостами.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. What are hypervisors. URL: https://www.ibm.com/topics/hypervisors (дата звернення: 15.12.2023).

2. What is high availability. URL: https://www.redhat.com/en/topics/linux/what-is-high-availability (дата звернення: 14.12.2023).

3. Тимощук В.Д., Чех Т.П., Фіялка А.І., Луцик Н.С. Методи віртуалізації в кластерах високої доступності. Матеріали XI науково-технічної конференції "Інформаційні моделі, системи та технології" Тернопільського національного технічного університету імені Івана Пулюя (Тернопіль, 13-14 грудня 2023 року). Тернопіль: ТНТУ. 2023. С. 186.

4. VMware vSphere Documentation. URL: https://docs.vmware.com/en/VMware-vSphere/index.html (дата звернення: 14.12.2023).

5. Тимощук В.Д., Тимощук Д.І. Віртуалізація в центрах обробки даних – аспекти відмовостійкості. Матеріали X науково-технічної конференції "Інформаційні моделі, системи та технології" Тернопільського національного технічного університету імені Івана Пулюя (Тернопіль, 7-8 грудня 2022 року). Тернопіль: ТНТУ. 2022. С. 95.

6. Hyper-V Technology Overview. [URL: https://learn.microsoft.com/en-us/windows-server/virtualization/hyper-v/hyper-v-technology-overview (дата звернення: 14.12.2023).

7. Failover Clustering in Windows Server. . URL: https://learn.microsoft.com/en-us/windows-server/failover-clustering/failover-clustering-overview (дата звернення: 14.12.2023).

8. Citrix Hypervisor. URL: https://docs.xenserver.com/en-us/citrix-hypervisor/ (дата звернення: 14.12.2023).

9. Kernel Virtual Machine. URL: https://linux-kvm.org/page/Main_Page (дата звернення: 14.12.2023).

10. Oracle Linux KVM and Virtualization Manager. URL: https://www.oracle.com/a/ocom/docs/oracle-linux-virtualization-manager-ds-final.pdf (дата звернення: 14.12.2023).

11. Чех Т.П., Тимощук В.Д., Кітчак Н.Ю., Луцик Н.С. Застосування гіпервізора KVM в кластерах високої доступності. Матеріали V міжнародної науково-практичної конференції «Scientific practice: modern and classical research methods» (22 грудня 2023 року). Бостон, США. 2023. С. 234

12. Details About Hardware Virtualization. URL: https://docs.oracle.com/en/virtualization/virtualbox/6.0/admin/hwvirt-details.html (дата звернення: 14.12.2023).

13. QEMU documentation. URL: https://www.qemu.org/docs/master/ (дата звернення: 14.12.2023).

14. oVirt User Documentation. URL: https://www.ovirt.org/documentation/ (дата звернення: 14.12.2023).

15. TrueNAS CORE tutorials. URL: https://www.truenas.com/docs/core /coretutorials/ (дата звернення: 14.12.2023).

16. Лупенко С.А., Луцик Н.С., Луцків А.М., Осухівська Г.М., Тиш Є.В. Методичні вказівки до виконання кваліфікаційної роботи магістра для студентів спеціальності 123 «Комп'ютерна інженерія» другого (магістерського) рівня вищої освіти усіх форм навчання. Тернопіль, ТНТУ. 2021. 34 с.

17. Закон України «Про охорону навколишнього природного середовища» №1264-XII. URL: https://zakon.rada.gov.ua/laws/ show/1264-12 (дата звернення: 14.12.2023).

18. Постанова Кабінету Міністрів України «Про затвердження Положення про державну систему моніторингу довкілля» №391-98-п. URL: https://zakon.rada.gov.ua/laws/show/391-98 (дата звернення: 14.12.2023).

19. Стручок В.С. Техноекологія та цивільна безпека. Частина «Цивільна безпека». Навчальний посібник. Тернопіль: ТНТУ. 2022. 150 с.

ДОДАТКИ

Додаток А Тези конференцій

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ПУЛЮЯ

М А Т Е Р І А Л И

XI НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

«ІНФОРМАЦІЙНІ МОДЕЛІ, СИСТЕМИ ТА ТЕХНОЛОГІЇ»

13-14 грудня 2023 року

ТЕРНОПІЛЬ
2023

Б.С.Таранін, О.Р.Цебрій
РОЗРОБКА СКРИПТУ ДЛЯ ІНТЕГРАЦІЇ СКЛАДОВИХ ЕЛЕМЕНТІВ ІНЕРЦІЙНОЇ СИСТЕМИ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЙ «ARDUPILOT»
B.S.Taranin, O.R.Tsebriy
DEVELOPMENT OF A SCRIPT FOR INTEGRATION COMPONENT ELEMENTS OF THE INERTIAL SYSTEM USING «ARDUPILOT» TECHNOLOGIES 182

М.В. Тененський
ПОРІВНЯННЯ БАЗ ДАНИХ MONODB ТА POSTGRESQL В КОНЕКСТІ РОЗРОБКИ СУЧАСНИХ ВЕБ-ЗАСТОСУНКІВ
M.V.Tenenskyi
COMPARISON OF MONGODB AND POSTGRESQL DATABASES IN THE CONTEXT OF MODER WEB APPLICATIONS DEVELOPMENT 184

В. Тимощук, Т. Чех, А. Фіялка, Н. Луцик
МЕТОДИ ВІРТУАЛІЗАЦІЇ В КЛАСТЕРАХ ВИСОКОЇ ДОСТУПНОСТІ
V. Tymoshchuk, T. Chekh, A. Fiialka, N. Lutsyk
METHODS OF VIRTUALIZATION IN HIGH AVAILABILITY CLUSTERS 186

В. Тимощук, В. Василишин, І. Мудрий, Н. Луцик
ОГЛЯД ТА ПОРІВНЯННЯ ПРОТОКОЛІВ ПЕРЕДАЧІ ІНФОРМАЦІЇ В ІОТ
V. Tymoshchuk, V. Vasylyshyn, I. Mudryi , N. Lutsyk
OVERVIEW AND COMPARISON OF INFORMATION TRANSFER PROTOCOLS IN IOT 188

Ткачук Р.М., Ткачук Р.А.
ЗАБЕЗПЕЧЕННЯ ІНДИВІДУАЛЬНОГО ПІДБОРУ КЛАПАНІВ ДЛЯ ВИВОДУ ВНУТРІШНЬООЧНОЇ РІДИНИ ПРИ ЛІКУВАННІ ГЛАУКОМИ
Tkachuk R.M., Tkachuk R.A.
PROVISION OF INDIVIDUAL SELECTION OF VALVES FOR THE REMOVAL OF INTRAOCULAR FLUID IN THE TREATMENT OF GLAUCOMA 189

Д. А. Урбан
СУЧАСНІ МОДЕЛІ ТА АЛГОРИТМИ ДЛЯ АВТОМАТИЗОВАНОГО АНАЛІЗУ ТА ПРОГНОЗУВАННЯ ДИНАМІКИ СОЦІАЛЬНИХ МЕДІА
D. A. Urban
MODERN MODELS AND ALGORITHMS FOR AUTOMATED ANALYSIS AND FORECASTING OF SOCIAL MEDIA DYNAMICS 190

Лілія Хвостівська, Назар Паламар, Сергій Сторож
ПРОГРАМНИЙ ЗАСІБ ВЕЙВЛЕТ-ВИЯВЛЕННЯ РАДІОСИГНАЛІВ В МАТЕРИНСЬКОМУ БАЗИСІ МЕКСИКАНСЬКОГО КАПЕЛЮХА
Liliia Khvostivska, Nazar Palamar, Serhii Storozh
SOFTWARE WAVELET DETECTION OF RADIO SIGNALS IN THE MEXICAN HAT MOTHER BASE 191

Д.Р. Чарковський, Н.Б. Стадник
МЕТОДИ ДЕТЕКТУВАННЯ ТЕКСТОВИХ ОБЛАСТЕЙ НА ЗОБРАЖЕННЯХ
D.R. Charkovkyi, N.B. Stadnyk
METHODS FOR DETECTION OF TEXT REGIONS IN IMAGES 192

Євгенія Тиш, Руслан Шалапай
ІЄРАРХІЧНА КЛАСТЕРИЗАЦІЯ ДЛЯ ВИЗНАЧЕННЯ СУКУПНОСТІ ФУНКЦІОНАЛЬНИХ ТА НЕФУНКЦІОНАЛЬНИХ ВИМОГ КОМП'ЮТЕРНИХ СИСТЕМ
Ievhenia Tysh, Ruslan Shalapay
HIERARCHICAL CLUSTERIZATION FOR DETERMINING FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS OF COMPUTER SYSTEMS 193

УДК 004.052.3
В. Тимощук, Т. Чех, А. Фіялка, Н. Луцик доктор філософії, доцент
(Тернопільський національний технічний університет імені Івана Пулюя)

МЕТОДИ ВІРТУАЛІЗАЦІЇ В КЛАСТЕРАХ ВИСОКОЇ ДОСТУПНОСТІ

V. Tymoshchuk, T. Chekh, A. Fiialka, N. Lutsyk Ph.D, Assoc. Prof.

METHODS OF VIRTUALIZATION IN HIGH AVAILABILITY CLUSTERS

Застосування віртуалізації для створення кластерів високої доступності забезпечує стійкість та надійність інформаційних систем. Віртуалізація дозволяє розділити фізичний апаратний ресурс на кілька віртуальних екземплярів, кожен з яких може працювати незалежно від інших. Цей підхід робить можливим більш ефективне та раціональне використання обладнання. У випадку виникнення проблеми з одним з віртуальних серверів, інші можуть продовжувати працювати, забезпечуючи неперервну роботу системи. Крім того, завдяки віртуалізації легко створювати резервні копії та відновлювати роботу системи у випадку відмови обладнання.

Створення кластерів високої доступності базується на об'єднанні декількох серверів у єдину систему, що забезпечує безперебійну роботу в разі відмови окремих компонентів. Такий підхід забезпечує підвищення доступності та надійності інформаційних систем. Кластеризація дозволяє розподілити завдання між серверами, уникнути ризику однієї точки відмови та забезпечити безперервну роботу системи.

Використання віртуалізації для створення кластерів високої доступності має ряд переваг. По-перше, це зменшує залежність від конкретного апаратного обладнання, оскільки віртуальні середовища можуть бути легко перенесені на інше обладнання у випадку несправності. По-друге, такий підхід дозволяє ефективніше використовувати ресурси серверів, оскільки при необхідності навантаження може бути розподілене між різними серверами в кластері.

Застосування віртуалізації для створення кластерів високої доступності передбачає використання різних платформ віртуалізації, таких як VMware vSphere, Microsoft Hyper-V, XEN та KVM, кожна з яких має свої унікальні можливості.

VMware vSphere є однією з провідних платформ віртуалізації. Вона надає широкі можливості управління віртуальними машинами та ресурсами серверів, забезпечуючи високу стабільність, безпеку та швидкодію. Ця платформа складається з декількох основних компонентів, що спільно працюють для створення та контролю віртуальних середовищ. Основними компонентами є гіпервізор VMware ESXi та vCenter Server. Функції High Availability та Fault Tolerance забезпечують автоматичне відновлення віртуальних машин у випадку відмови обладнання.

Microsoft Hyper-V є іншою популярною платформою віртуалізації, яка надає широкі можливості для створення віртуальних середовищ. Вона інтегрована з продуктами Microsoft, що дозволяє легко інтегрувати віртуальні машини з іншими сервісами та програмами. Microsoft Hyper-V також підтримує технологію Failover Cluster, що забезпечує високу доступність віртуальних машин.

XEN - це ще одна платформа віртуалізації, яка дозволяє створювати віртуальні сервери на базі різних операційних систем. Ця платформа відома своєю високою швидкодією та низькими вимогами до апаратного забезпечення. Рішення, побудовані на

базі XEN, такі як Citrix Hypervisor та XCP-ng Xen Hypervisor, також підтримують технологію High Availability, що дозволяє створювати стійкі до відмов системи.

KVM (Kernel-based Virtual Machine) - це технологія віртуалізації, що використовує ядро операційної системи Linux для управління віртуальними машинами. KVM забезпечує високу продуктивність і можливість роботи на різному обладнанні. Продукти, такі як Oracle Linux Virtualization Manager та Red Hat Enterprise Virtualization, використовують KVM з підтримкою технології High Availability для створення високодоступних кластерів.

Усі ці платформи дозволяють створювати стійкі до відмов кластери, зменшуючи залежність від конкретного обладнання та ефективно розподіляючи ресурси між віртуальними середовищами. Однак, використання віртуалізації для кластерів високої доступності потребує уважного управління безпекою. Порушення безпеки в одному з серверів може вплинути на весь кластер та віртуальні машини. Тому важливо розробляти ефективні стратегії забезпечення безпеки та відновлення системи у разі виявлення загроз.

Використання віртуалізації для створення кластерів високої доступності є ключовим елементом у розвитку сучасних інформаційних технологій. Вибір оптимальної платформи віртуалізації відповідно до потреб і вимог конкретного проекту є вирішальним для забезпечення безперебійної та надійної роботи інформаційних систем.

ΛΌΓΟΣ

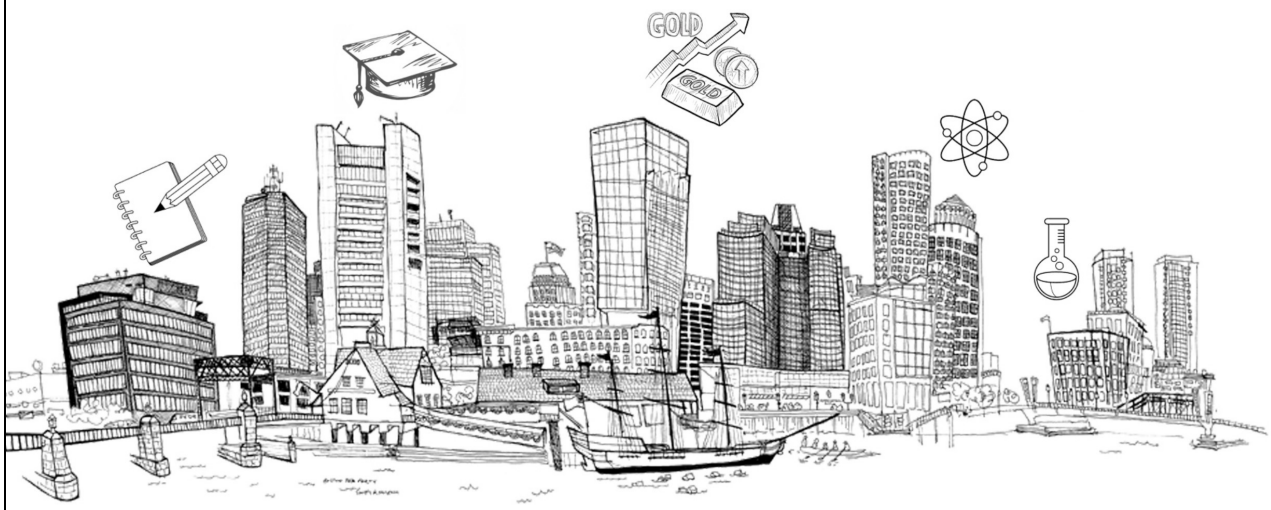
COLLECTION OF SCIENTIFIC PAPERS

THE ART OF SCIENTIFIC MIND

WITH PROCEEDINGS OF THE V INTERNATIONAL SCIENTIFIC AND PRACTICAL CONFERENCE

SCIENTIFIC PRACTICE: MODERN AND CLASSICAL RESEARCH METHODS

DECEMBER 22, 2023 • BOSTON, USA



ISBN 979-8-89217-818-1 (PDF) ISBN 978-617-8126-70-4 DOI 10.36074/logos-22.12.2023

ASSESSING AI-DRIVEN MACHINE TRANSLATION PERFORMANCE FROM ENGLISH INTO UKRAINIAN USING BLEU METRICS FOR IOT PRODUCT LOCALIZATION
Kostina O., Horbovyy A. 205

GLOBAL TRENDS IN THE DEVELOPMENT OF CLOUD SOLUTIONS AND TECHNOLOGIES
Kushchov O. 213

HEALING WITH ALGORITHMS: HOW MACHINE LEARNING TRANSFORMS HEALTHCARE
Sliusarenko T. 218

MACHINE LEARNING - METHOD OR SOLUTION?
Scientific research group:
Miller T., Kozlovska P., Łobodzińska A., Lewita K., Żejmo J., Kaczanowska O. 220

MODERN MEANS OF AUTOMATION OF THE SUBWAY OPERATION
Panchenko A.A., Frolov S.V., Golub G.M. 226

АЛГОРИТМІЧНЕ І МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ ЛАЗЕРНОГО КОРЕЛЯЦІЙНОГО СПЕКТРОМЕТРА ДЛЯ ДОСЛІДЖЕНЬ НАНООБ'ЄКТІВ
Яремик Р.Я. 228

ЗАСТОСУВАННЯ ГІПЕРВІЗОРА KVM В КЛАСТЕРАХ ВИСОКОЇ ДОСТУПНОСТІ
Чех Т.П., Тимощук В.Д., Кітчак Н.Ю. 234

МОДЕЛЬ ТА ЗАСОБИ ІНФОРМАЦІЙНОЇ СИСТЕМИ КЛАСИФІКАЦІЇ АВТОМОБІЛІВ
Басараб Ю.М., Гадьо І.В. 236

ПРО РОЗРОБЛЕННЯ WEB 3.0 ГРИ З ВИКОРИСТАННЯМ БЛОКЧЕЙНУ SOLANA
Мельников М.А., Гадьо І.В. 239

ПРОГНОЗУВАННЯ МЕДИЧНИХ ДАНИХ З ВИКОРИСТАННЯМ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ
Мамай Б.В., Гадьо І.В. 241

SECTION XXI.
TRANSPORT AND TRANSPORT TECHNOLOGIES

МОДЕЛЬ ВИКОРИСТАННЯ БЕЗПІЛОТНИХ ПОВІТРЯНИХ СУДЕН В АЕРОПОРТАХ
Суркова К.В., Данилко О.Г. 243

DOI 10.36074/logos-22.12.2023.061

ЗАСТОСУВАННЯ ГІПЕРВІЗОРА KVM В КЛАСТЕРАХ ВИСОКОЇ ДОСТУПНОСТІ

Чех Тарас Павлович
здобувач вищої освіти
факультету комп'ютерно-інформаційних систем і програмної інженерії
Тернопільський національний технічний університет імені Івана Пулюя

ORCID ID: 0009-0007-2858-9434 **Тимощук Віталій Дмитрович**
здобувач вищої освіти
факультету прикладних інформаційних технологій та електроінженерії
Тернопільський національний технічний університет імені Івана Пулюя

Кітчак Назар Юрійович
здобувач вищої освіти
факультету комп'ютерно-інформаційних систем і програмної інженерії
Тернопільський національний технічний університет імені Івана Пулюя

НАУКОВИЙ КЕРІВНИК:

ORCID ID: 0000-0002-0361-6471 **Луцик Надія Степанівна**
доктор філософії, доцент кафедри комп'ютерних систем та мереж
Тернопільський національний технічний університет імені Івана Пулюя

УКРАЇНА

Гіпервізор KVM (Kernel-based Virtual Machine) є одним із провідних інструментів у сфері віртуалізації, який надає можливість створення та управління віртуальними машинами на базі Linux [1].

Кластери високої доступності (High Availability) - це складні системи, спрямовані на забезпечення безперервної роботи сервісів і додатків у випадку виникнення збоїв апаратних чи програмних компонентів [2]. Використання гіпервізора KVM у таких кластерах дозволяє досягти високого рівня доступності, використовуючи переваги віртуалізації для забезпечення незалежності сервісів від конкретного апаратного забезпечення.

Однією з ключових переваг застосування KVM у кластерах HA є можливість міграції віртуальних машин між фізичними серверами без перерви в роботі сервісів. Ця функція, відома як Live Migration, дозволяє автоматично переміщати робочі навантаження з одного сервера на інший для збільшення ефективності використання ресурсів та уникнення відмови системи. Крім того, кластери високої доступності забезпечують можливість балансування навантаження між серверами кластера, що сприяє оптимальному розподілу ресурсів і підвищує відмовостійкість системи в цілому. Функція автоматичного відновлення роботи віртуальних машин у разі збою хоста кластера дозволяє автоматично перезапускати віртуальні машини на іншому працюючому хості у кластері.

Ще однією важливою характеристикою KVM є його відкритість та гнучкість налаштування. Будучи вбудованим у ядро Linux, KVM використовує відкриті стандарти і протоколи, що робить його досить гнучким у поєднанні з іншими інструментами та технологіями. Це дає можливість інтегрувати KVM у

різноманітні стеки програмного забезпечення, використовуючи його для створення стійких та масштабованих інфраструктур.

На сьогоднішній день існує кілька платформ віртуалізації, які базуються на гіпервізорі KVM і надають різноманітні функції для вирішення потреб різних користувачів. Ось кілька з них: Proxmox Virtual Environment (Proxmox VE), Oracle Linux Virtualization Manager (OLVM) та Red Hat Enterprise Virtualization (RHEV). Ці платформи віртуалізації надають різноманітні можливості для створення та управління віртуальними середовищами. Вони дозволяють користувачам створювати, налаштовувати та керувати віртуальними машинами, забезпечуючи надійність, безпеку та масштабованість.

Рішення з використанням кластерів віртуалізації на базі KVM мають широкий спектр застосувань у сучасних інформаційних технологіях. Компанії, що надають послуги хмарних обчислень, використовують KVM для побудови власних інфраструктурних сервісів. Наприклад, OpenStack, популярна платформа для створення хмарних обчислень, використовує гіпервізор KVM для створення та управління віртуальними машинами [3]. Також Google Cloud використовує модифіковану версію KVM та власну реалізацію монітору віртуальних машин в просторі користувача відмінну від QEMU. Ці модифікації та вдосконалення допомагають Google Cloud забезпечувати високий рівень безпеки та ефективності використання ресурсів для своїх клієнтів у хмарному середовищі [4].

Багато компаній використовують кластери віртуалізації на основі KVM для побудови власних приватних хмар для віртуалізації своїх корпоративних ресурсів. Наприклад, банки, телекомунікаційні компанії або медичні установи створюють віртуальні середовища для зберігання конфіденційних даних та запуску критичних застосунків. Хостингові компанії, що надають віртуальне хостингове середовище або послуги для створення вебсайтів, використовують KVM для створення та управління віртуальними серверами клієнтів. Команди розробників та тестувальників використовують віртуалізацію KVM для створення ізольованих середовищ для тестування програмного забезпечення або розгортання різних конфігурацій для розробки. Компанії, що працюють з великим обсягом даних або мають високі вимоги до обчислювальних ресурсів, використовують кластери віртуалізації на базі KVM для ефективного управління ресурсами та їх оптимального розподілу.

Отже, гіпервізор KVM є потужним інструментом для створення віртуальних середовищ у кластерах високої доступності. Його можливості забезпечують гнучкість, масштабованість та надійність віртуальних інфраструктур.

Список використаних джерел:

[1] Kernel Virtual Machine [Електронний ресурс]. — URL: https://linux-kvm.org/page/Main_Page (дата звернення: 14.12.2023).

[2] What is high availability [Електронний ресурс]. — URL: https://www.redhat.com/en/topics/linux/what-is-high-availability (дата звернення: 14.12.2023).

[3] OpenStack KVM [Електронний ресурс]. — URL: https://docs.openstack.org/mitaka/config-reference/compute/hypervisor-kvm.html (дата звернення: 14.12.2023).

[4] KVM hypervisor at Google Cloud [Електронний ресурс]. — URL: https://cloud.google.com/blog/products/gcp/7-ways-we-harden-our-kvm-hypervisor-at-google-cloud-security-in-plaintext (дата звернення: 14.12.2023).

Додаток Б Блок-схема алгоритму роботи кластера