ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖНЕРІЇ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Дослідження сучасних технологій віртуалізації та реалізація їх за допомогою гіпервізора 1 типу "Proxmox"»

на здобуття освітнього ступеня магістра
зі спеціальності 123 Комп'ютерні системи та мережі
(код, найменування спеціальності)
освітньо-професійної програми Комп'ютерні системи та мережі
(назва)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ Дмитро ВАСИЛЕНКО
(підпис) *Ім'я, ПРІЗВИЩЕ здобувача*

Виконав:
здобувач вищої освіти Дмитро ВАСИЛЕНКО
група КСДМ-61

Керівник: В'ячеслав ЧЕРЕВИК
науковий ступінь, вчене звання к.т.н., доцент

Рецензент:
науковий ступінь, вчене звання _____
Ім'я, ПРІЗВИЩЕ

Київ 2023

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут Інформаційних технологій

Кафедра Комп'ютерної інженерії

Ступінь вищої освіти Магістр

Спеціальність 123 «Комп'ютерна інженерія»

Освітньо-професійна програма Комп'ютерні системи та мережі

ЗАТВЕРДЖУЮ
Завідувач кафедрою Комп'ютерної інженерії
_____ Наталія ЛАЩЕВСЬКА
«_____» _____ 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Василенко Дмитру Євгеновичу
(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Дослідження сучасних технологій віртуалізації та реалізація їх за допомогою гіпервізора 1 типу "Proxmox"

керівник кваліфікаційної роботи В'ячеслав ЧЕРЕВИК к.т.н., професор,
(Ім'я, ПРІЗВИЩЕ науковий ступінь, вчене звання)

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «19» жовтня 2023р. №145

2. Строк подання кваліфікаційної роботи «28» грудня 2023р.

3. Вихідні дані до кваліфікаційної роботи: Поняття віртуалізації, сучасні технології віртуалізації, способи реалізації віртуаалізації, гіпервізор 1 типу Proxmox.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Дослідження технологій віртуалізації
2. Реалізація технології різними способами
3. Реалізація технологій за допомогою гіпервізора 1 типу Proxmox

5. Перелік графічного матеріалу: *презентація*

1. Мета, об'єкт та предмет дослідження
2. Актуальність теми,
3. Технологія віртуалізації
4. Віртуальна машина

5. Гіпервізор 1 та 2 типу
6. Віртуалізація серверної частини
7. Proxmox
8. Кластеризація
9. Спільне схивище ZFS
10. Реплікація
11. Висока доступність Proxmox
12. Висновки

6. Дата видачі завдання «19» жовтня 2023 р._____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Аналіз наявної науково-технічної літератури	19.10-24.10.23	Виконано
2	Дослідження технологій віртуалізіції.	24.10-30.11.23	Виконано
3	Дослідження реалізацій сучасних технологій віртуалізіції	30.11-02.12.23	Виконано
4	Обробка матеріалу	02.12-04.12.23	Виконано
5	Висновки за результатами аналізу	04.12-06.12.23	Виконано
6	Розробка теоретичної частини	06.12-15.12.23	Виконано
7	Оформлення роботи: вступ, висновки, реферат	15.12-26.12.23	Виконано
8	Розробка демонстраційних матеріалів	26.12-27.12.23	Виконано

Здобувач вищої освіти _____ Дмитро ВАСИЛЕНКО
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

Керівник
кваліфікаційної роботи _____ В'ячеслав ЧЕРЕВИК
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 83 стор., 29 рис., 50 джерел.

Мета роботи – провести глибокий аналіз сучасних технологій віртуалізації, зосереджуючись на гіпервізорі 1 типу "Proxmox".

Об'єкт дослідження – реалізація сучасних технологій віртуалізації за допомогою програмних продуктів WMware, VM VirtualBox та Proxmox.

Предмет дослідження – ESXi та vSphere від компанії WMware , Oracle VM VirtualBox та Proxmox.

Короткий зміст роботи: Для створення серверної компоненти з кластерами для управління нею в роботі досліджено технології віртуалізації та застосовано їх на практиці. Робота включає в себе розгляд теоретичних основ віртуалізації, методів розробки та детальний виклад експериментального дослідження з використанням технологій віртуалізації. Результати цього експерименту доводять, що технологія ефективна та позитивно впливає на безпеку та ефективність корпоративної інфраструктури. Ці результати підсумовуються у висновках і визначають напрями подальших досліджень у цій галузі.

КЛЮЧОВІ СЛОВА: ВІРТУАЛЬНА МАШИНА, ПОВНА ВІРТУАЛІЗАЦІЯ, АПАРАТНА ВІРТУАЛІЗАЦІЯ, ПАРАВІРТУАЛІЗАЦІЯ, ВІРТУАЛІЗАЦІЯ РОБОЧОГО СТОЛУ, ГІБРИДНИЙ ГІПЕРВІЗОР, ВІРТУАЛІЗАЦІЯ СЕРВЕРУ, ORACLE VM VIRTUAL BOX ,VMWARE WORKSTATION, VMWARE ESXI , PROXMOX.

ABSTRACT

The text part of the qualification work for obtaining a master's degree: 83 pages 29 figures, 31 sources.

The purpose of the work is to conduct a deep analysis of modern virtualization technologies, focusing on type 1 hypervisor.

The object of the study is the implementation of modern virtualization technologies using WMware, VM VirtualBox and Proxmox software products.

The subject of the research is ESXi and vSphere from WMware, Oracle VM VirtualBox and Proxmox.

Summary of work: In order to create a server component with clusters for its management, virtualization technologies were researched and applied in practice. The work includes consideration of the theoretical foundations of virtualization, development methods, and a detailed account of experimental research using virtualization technologies. The results of this experiment prove that the technology is effective and has a positive effect on the security and efficiency of the corporate infrastructure. These results are summarized in the conclusions and determine the direction of further research in this field.

KEYWORDS: VIRTUAL MACHINE, FULL VIRTUALIZATION, HARDWARE VIRTUALIZATION, PARAVIRTUALIZATION, DESKTOP VIRTUALIZATION, HYBRID HYPERVISOR, SERVER VIRTUALIZATION, ORACLE VM VIRTUAL BOX ,VMWARE WORKSTATION, VMWARE ESXI , PROXM OX.

ЗМІСТ

ВСТУП

В умовах стрімкого розвитку технологій та широкого застосування різних технологій та обладнення, забезпечення зручного та ефективного методу адміністрування цих систем стає необхідністю для сучасних організацій. Методика віртуалізації серверного обладнення виходить за межі звичайного, пропонуючи повноцінне рішення для забезпечення безпеки та оптимального використання IT-ресурсів.

Основні завдання віртуалізації включають:

– Віртуалізація дозволяє об'єднувати різні обчислювальні ресурси (сервери, зберігання) на одному фізичному сервері. Це дозволяє ефективніше використовувати апаратні ресурси та зменшити кількість необхідного обладнання;

– Забезпеченя ізоляції віртуальних машин одна від одної. Це означає, що проблеми, що виникають на одній VM, не повинні впливати на інші, що забезпечує вищий рівень надійності та безпеки;

– Віртуалізація дозволяє швидко створювати та розгортати віртуальні машини, швидко змінювати розмір та обсяг віртуальних ресурсів залежно від потреб. Це полегшує процес введення нових обчислювальних ресурсів у виртуальне середовище та робить її більш еластичною.

Окрім технічного рішення, цей підхід також спрямований на те, щоб зробити компанію більш гнучкою та конкурентоспроможною в технологічному середовищі, що постійно змінюється. Інтеграція систем віртуаліації може запропонувати говий стандарт для адміністрування.

РОЗДІЛ 1 ДОСЛІДЖЕННЯ ТЕХНОЛОГІЙ ВІРТУАЛІЗАЦІЇ

1.1 Поняття віртуалізіції

Термін віртуалізація означає створення віртуального або змодельованого обчислювального середовища, а не реального. Апаратне забезпечення, операційні системи, пристрої зберігання та інші версії програмного та апаратного забезпечення, розроблені комп'ютером, часто інтегруються у віртуалізацію. Це дає змогу організаціям розділити один комп'ютер або сервер на декілька віртуальних машин. Кожна віртуальна машина може запускати власну операційну систему або програму та спільно використовувати ресурси одного хост-комп'ютера.

Завдяки оптимізації масштабованості та робочих навантажень віртуалізація мінімізує кількість серверів, зменшує споживання енергії та знижує витрати на обслуговування.

1.2 Віртуальна машина

Віртуальні машини — це комп'ютерна модель, створена шляхом віртуалізації обчислювальних ресурсів, таких як процесори, блоки оперативної пам'яті, пристрої зберігання даних, а також введення та виведення інформації. На відміну від програми емуляції для конкретного пристрою, віртуальні машини можуть емулювати практично будь-яку фізичну машину або середовище виконання для програми.

У якийсь момент віртуальна машина була визначена як «дійсна окрема версія фактичної машини». Віртуальні машини в сучасний час можуть не мати апаратного аналога. Віртуальна машина може моделювати реальні або абстрактні обчислювальні машини на основі моделювання набору інструкцій їхнього віртуального ЦП. Процесор віртуальної машини та обраний набір інструкцій центрального процесора нерозрізнені під час імітації реального комп'ютера.

Існує два основних типи віртуальних машин, які відрізняються за використанням і сумісністю з реальним обладнанням:

– апаратні (системні) — це віртуалізовані машини, які емулюють всю апаратну платформу і таким чином віртуальна машина має смогу підтримують усю роботу операційної системи;

– прикладні — це віртуальні машини, які можуть запускати лише додатки (прикладні програми) .

Системні віртуальні машини можуть розділяти апаратні ресурси фізичної машини на кілька копій, кожна з яких може мати власну операційну систему.

Основні переваги використання системних віртуальних машин:

– комп’ютер може мати декілька операційних систем, що працюють у різний час, зберігаючи сувору ізоляцію одна від одної;

– розширений набір машинних інструкцій може надаватися системними віртуальними машинами, оскільки інструкції для процесора віртуальної машини при моделюванні абстрактного комп’ютера можуть відрізнятися;

– широкий контроль програми;

– легкі зміни та ремонт.

Через опосередкований доступ апаратного забезпечення системна віртуальна машина має менше ефективності порівняно зі своїм реальним аналогом.

Зазвичай для обладнання серверів використовують різні віртуальні машини, кожна зі своєю власною операційною системою. Для роботи різних служб на різних віртуальних машинах потрібні окремі машини, але їх можна підключити через одну фізичну машину та використовувати ресурси, доступні в апаратному забезпеченні, без шкоди для продуктивності.

Операційна система включає звичайні програми, які виконуються прикладними віртуальними машинами. Як правило, вони створюються на початку програми та видаляються після її завершення. Мета полягає в тому, щоб забезпечити середовище програмування, яке дозволяє абстрагуватися від конкретного апаратного забезпечення та операційної системи, на якій виконується програма.

На відміну від прикладних віртуальних машин, системні віртуальні машини зазвичай обмежуються абстракцією низького рівня. Що означає високий рівень абстракція в прикладних віртуальних машин. Компіляція на льоту використовується для збільшення швидкості виконання на сучасних віртуальних машинах додатків, реалізованих за допомогою інтерпретаторів.

1.3 Апаратна віртуалізація

Віртуалізація апаратного забезпечення передбачає віртуалізацію комп'ютерів як повних апаратних платформ, логічних абстракцій від їхніх компонентів або функцій різних операційних систем. Представляючи абстрактну обчислювальну платформу, віртуалізація приховує від користувачів фізичні властивості обчислювальних платформ. Спочатку програмне забезпечення віртуалізації було відоме як «драйвер», але пізніше його замінили термінами «гіпервізор» або «монітор віртуальної машини».

Віртуалізація апаратної платформи досягається за допомогою програмного забезпечення хоста, яке створює віртуальні машини для свого гостьового програмного забезпечення. Багато хостів пропонують повноцінні операційні системи, що дозволяє користувачам використовувати гостьове програмне забезпечення самостійно. З кількома важливими застереженнями гостьове програмне забезпечення працює так, ніби воно працює безпосередньо на фізичному обладнанні. Контроль фізичних системних ресурсів, включаючи доступ до мережі, дисплей, клавіатуру та дискове сховище, як правило, більш обмежений, ніж керування хостом, процесором або системною пам'яттю. Хости віртуалізації можуть обмежувати доступ до певних периферійних пристроїв або обмежувати можливості пристрою набором спеціальних функцій, залежно від політик доступу до обладнання. Віртуалізація може призвести до зниження продуктивності, що спричинено ресурсами, необхідними для роботи гіпервізора, і продуктивністю віртуальних машин порівняно з фізичними.

Переваги апаратної віртуалізації:

– віртуальні машини ізолюють одну операційну систему від іншої та від хост-системи. Це дозволяє запускати різні ОС і програми на одному фізичному сервері без взаємовпливу;

– системні віртуальні машини дозволяють використовувати фізичні ресурси (процесор, пам'ять, сховище) більш ефективно, оскільки кілька віртуальних машин може спільно використовувати ресурси одного сервера;

– віртуальні машини можна легко переносити з одного фізичного сервера на інший, що дозволяє оптимізувати використання ресурсів і управляти навантаженням;

– завдяки тому, що віртуальні машини легко переносяться, їх можна використовувати в сценаріях аварійного відновлення, не турбуючись про вплив збоїв живлення та відновлення;

– віртуалізація дозволяє ефективно розподіляти ресурси між віртуальними машинами в залежності від їх потреб, що дозволяє оптимізувати використання обладнання;

– моніторинг і перевірка віртуальної машини є більш простими із зовнішнього місця, ніж з фізичного місця, а її конфігурація більш універсальна;

– віртуальні машини можуть бути легко масштабовані вгору або вниз в залежності від потреб користувача або бізнес-задач.

Тим не менш, коли кілька віртуальних машин працюють одночасно на одному фізичному хості, кожна віртуальна машина може демонструвати непередбачувану та змінну продуктивність, яка значною мірою залежить від навантаження системи з інших віртуальних комп’ютерів. Щоб вирішити цю проблему, можна використати відповідні методи встановлення для тимчасової ізоляції віртуальних машин. Апаратна віртуалізація також зменшує витрати на технічне обслуговування паравіртуалізованих систем за рахунок зменшення змін гостьової операційної системи. Апаратна віртуалізація вимагає явної підтримки в центральному ЦП, яка присутня не у всіх процесорах x86/x64. Як було виявлено в 2006 році, підтримка апаратного забезпечення першого покоління 32- і 64-розрядних над віртуалізацією програмного забезпечення часто не забезпечувала

переваги продуктивності. Окрім цього апаратна віртуалізіція ще декілька недоліків:

– віртуалізація завжди призводить до деякої втрати продуктивності через додатковий шар віртуалізації та управління ресурсами;

– деякі застосунки, зокрема ті, які вимагають великої продуктивності графіки (наприклад, ігри або графічні додатки), можуть відчути обмеження у віртуальному середовищі;

– хоча велика частина віртуалізаційного програмного забезпечення є безпечною, вона може мати вразливості, які можуть бути використані для атак.

– деяке обладнання може не підтримувати віртуалізацію або працювати менш ефективно в середовищі віртуалізації;

– налаштування та управління віртуальними машинами може бути складним завданням і вимагати додаткового часу та експертизи;

– віртуальні машини можуть вимагати ліцензій для віртуалізаційного програмного забезпечення, а також можуть потребувати додаткових витрат на обслуговування та підтримку;

– хоча віртуалізація може допомагати ефективніше використовувати ресурси, вона все ще потребує додаткових обчислювальних ресурсів для управління віртуальними машинами;

Загалом, багато з цих недоліків можна вирішити або пом'якшити за допомогою правильної конфігурації та управління віртуальними машинами.

1.4 Повна віртуалізація

Повна віртуалізація вимагає зіставлення всіх важливих апаратних функцій (тобто набору інструкцій, операцій вводу/виводу, переривань і доступу до пам’яті) на одну або кілька віртуальних машин, які також призначені для роботи на віртуальній машині, а не на фізичній віртуалізованій голій машині. Операційні системи можна використовувати в будь-якому середовищі віртуальної машини, яке включає все програмне забезпечення, яке може працювати на необробленому

обладнанні. Найбільш очевидним показником повної віртуалізації є те, чи може автономна операційна система ефективно функціонувати в просторі віртуальної машини. Для ефективного функціонування повної віртуалізації потрібна відповідна комбінація апаратного та програмного забезпечення. Гіпервізори та супероперацийні системи, які працюють на рівні привілеїв, вищому за ОС, зазвичай називають віртуалізацією 1-го типу або повною віртуалізацією. Однією з основних перешкод у повній віртуалізації є захоплення та моделювання привілейованих операцій, таких як інструкції введення/виведення. Кожна операція віртуальної машини має тривати у віртуалізованій системі, щоб підтримувати стан інших віртуальних машин, програм керування чи обладнання. Деякі інструкції на машинах можуть бути виконані безпосередньо за допомогою апаратного забезпечення, оскільки їхні ефекти повністю містяться в компонентах, керованих драйвером, таких як розташування пам’яті та арифметичні регістри. Віртуальна машина не може бути виконана безпосередньо іншими інструкціями, тому їх потрібно захопити та змоделювати. Це проблематично. Ці інструкції відповідають за доступ або вплив на інформацію про стан поза віртуальною машиною.Повна віртуалізація виявилася дуже успішною для:

– спільне використання комп'ютерної системи між кількома користувачами;

– ізоляція користувачів і керуючих програм;

– підвищеня надійності, безпеки та продуктивністю завдяки розробці нових методів емуляції апаратного забезпечення.

1.5 Паравіртуалізація

Паравіртуалізація — це метод віртуалізації в обчислювальній техніці, який забезпечує, хоча й не ідентичний базовому програмному апаратному інтерфейсу але схожий.

Метою нового інтерфейсу є мінімізація часу виконання гостьової системи, дозволяючи їй виконувати завдання, які складніше виконувати у віртуальному середовищі. На віртуальний домен, де продуктивність виконання нижча, може

вплинути використання спеціально створених «гачків», наданих парвіртуалізацією. Використовуючи добре спроектовану паравіртуалізовану платформу, стає легше контролювати віртуальні машини, переносячи критичні завдання з віртуального домену на хост-домен, що, у свою чергу, зменшує погіршення загальної продуктивності машини.

Для паравіртуалізації необхідно перенести гостьову операційну систему на API, а звичайний дистрибутив ОС, який не підтримує паравіртуалізацію, не можна використовувати на змішаних моніторах віртуальних машин. Хоча операційна система не може бути змінена, паравіртуалізація може запропонувати переваги продуктивності завдяки використанню певних компонентів. Щоб продемонструвати, проект Xen Windows GPLPV пропонує колекцію драйверів пристроїв, які дозволяють паравіртуалізацію та ліцензовані під GPL для встановлення на віртуальних гостьових пристроях, що працюють під керуванням операційної системи Microsoft Windows (гіпервізор XEN). Такі програми зазвичай доступні через середовище паравіртуального машинного інтерфейсу. Забезпечується сумісність під час виконання з різними моделями алгоритмів шифрування, що дозволяє бездоганно інтегрувати його в паравіртуальну структуру. Основними перевагами та недоліками паравіртуалізацією є:

– ефективніше використовування ресурсів фізичного сервера через активну співпрацю між гіпервізором і гостьовою системою;

– у порівнянні з повною віртуалізацією, паравіртуалізація може зменшити витрати на обчислювальні ресурси та енергопотребу;

– гістьові операційні системи, які розуміють свою віртуалізовану природу, можуть досягти кращої продуктивності завдяки співпраці з гіпервізором;

– паравіртуалізація може бути корисною в ситуаціях, де використовуються спеціальні операційні системи або системи, що підтримують паравіртуалізацію, такі як Xen;

– гостьові операційні системи повинні бути модифіковані для підтримки паравіртуалізації. Це може вимагати додаткової роботи при впровадженні.

– хоча паравіртуалізація може покращити продуктивність в багатьох випадках, в окремих сценаріях вона може призвести до деякої втрати продуктивності;

– паравіртуалізація може вимагати, щоб гіпервізор та операційні системи були спеціально підтримані та модифіковані для підтримки.

1.6 Віртуалізація з апаратною підтримкою

Віртуалізація з апаратним забезпеченням — це тип віртуалізованої платформи, яка використовує можливості апаратного забезпечення, як правило, від головних процесорів, для досягнення ефективної повної віртуалізації. Використання повної віртуалізації дозволяє симулювати повне апаратне середовище або віртуальну машину, де немодифікована гостьова операційна система (використовуючи той самий набір інструкцій, що й головна машина) працює повністю ізольовано. Процесори x86 були оснащені апаратною віртуалізацією. еар. Віртуалізація з апаратним забезпеченням також відома як прискорена віртуалізація.

1.7 Віртуалізація робочого столу

Використовуючи віртуалізацію робочого столу, він відокремлює фізичний клієнтський пристрій від реального робочого середовища та пов’язаного прикладного програмного забезпечення. Це відоме як розділення робочого столу.

Завдяки інтеграції віртуалізації робочого столу з віртуалізацією додатків і системами керування профілями користувачів, які зазвичай називають віртуалізованими користувачами, користувачі можуть мати повний контроль над своїм робочим середовищем. У цьому режимі всі складові частини робочого столу віртуалізуються, що забезпечує дуже гнучку та набагато безпечнішу модель доставки робочого столу. Крім того, ця техніка підтримує більш комплексний підхід до аварійного відновлення робочого столу, оскільки всі частини

зберігаються в центрі обробки даних і захищені традиційними системами підтримки резервного копіювання. Наявність компонентів під час входу з іншого пристрою спрощує відновлення в разі викрадення пристрою чи обладнання користувача. Це забезпечує його надійність. Відсутність даних на пристрої користувача означає, що будь-яка важлива інформація з меншою ймовірністю буде доступна та зламана.

Реалізації віртуалізації робочого столу класифікуються відповідно до того, чи доступний віртуальний робочий стіл віддалено чи локально, чи є постійний доступ обов'язковим чи періодично запланованим, і чи підтримується він між сеансами. Ці параметри необов'язкові, але їх можна використовувати для конфігурації. Як правило, постачальники рішень для віртуалізації настільних комп'ютерів можуть об'єднувати локальні та віддалені впровадження в один продукт, щоб забезпечити відповідність конкретним вимогам. Стратегія доступу та розташування сервера визначають ступінь автономності, у якій виконуються функції клієнтського пристрою. Віртуалізація не є необхідною умовою для наявності дистанційного керування. Основна мета віртуалізації полягає в тому, щоб демонструвати автономні екземпляри декільком користувачам і наказувати стратегічну сегментацію хост-сервера та представлення в системі хостингу в певній точці її структури.

Впровадження віртуалізації для віддалених робочих столів підтримуються налаштуваннями клієнт-сервера. За допомогою протоколу віддаленого відображення користувач взаємодіє з програмами на локальному клієнтському пристрої, якщо він підключений до віддаленої операційної системи або спілкується з нею через мережу. Віддалена система зберігає всі програми та інформацію, одночасно передаючи дані з дисплея, клавіатури чи миші на локальний клієнтський пристрій, яким може бути персональний комп'ютер, гаджет із тонким клієнтом, планшет (або смартфон). Розміщення кількох екземплярів настільної операційної системи на серверній апаратній платформі гіпервізора є однією з поширених реалізацій цього підходу. Найновіша версія назва є інфраструктурою віртуальних робочих столів.

1.8 Віртуалізація додатків

Процес інкапсуляції комп'ютерних програм із операційної системи, у якій вони працюють, відомий як віртуалізація додатків. Інсталяція повністю віртуалізованої програми не виконується традиційними методами, але вона все одно працює за призначенням. Під час виконання програма поводиться так, ніби вона безпосередньо пов'язана з початковою операційною системою та всіма її ресурсами, але може бути окремою або обмеженою в її обробці.

Для повної віртуалізації програми необхідний особливий тип рівня віртуалізації. Середовище виконання програми зазвичай забезпечується операційною системою, а її замінюють рівні віртуалізації програми. Усі операції віртуалізованої програми перехоплюються цим рівнем, який потім спрямовує їх у віртуалізоване розташування, яке зазвичай містить лише один файл. Програма не може визначити, що вона звертається до віртуального ресурсу, а не лише до фактичного об'єкта. Використовуючи один файл для програми замість кількох файлів у системі, її без зусиль буде запустити на іншому комп'ютері.

Віртуалізація додатків дозволяє додаткам працювати в середовищах, які несумісні з рідними додатками. Програми Microsoft Windows можна запускати в Linux, наприклад за допомогою WINE.

Віртуалізація додатків може зменшити витрати на системну інтеграцію та адміністрування, забезпечуючи єдине базове програмне забезпечення для кількох комп'ютерів в організації. Операційна система та інші додатки захищені від поганого або помилкового коду через обмежену інтеграцію. Це важливо. Він може запропонувати захист пам'яті, можливості налагодження у стилі IDE та можливість виконувати погано написані програми, наприклад ті, які спрямовані на зберігання даних користувача в системі лише для читання. Це дозволяє одночасно запускати несумісні програми з мінімальним регресійним тестуванням один проти одного. Від'єднання програм від операційної системи вигідно з міркувань безпеки, оскільки відкриття однієї віртуалізованої програми не відкриває автоматично всю операційну систему.

Віртуалізація в програмах спрощує міграцію операційної системи. Програми можна зробити портативними, перемістивши їх на знімний носій або між комп’ютерами без встановлення. Сама по собі віртуальна машина споживає більше ресурсів, ніж віртуалізація програм.

1.9 Віртуалізація на рівні операційної системи

Схема віртуалізації операційної системи на рівні ОС допускає кілька екземплярів простору користувача, включаючи контейнери та зони. Програми на таких екземплярах можна порівняти з реальними комп'ютерами. Комп’ютерна програма на типовій операційній системі може отримувати доступ і аналізувати всі доступні ресурси, включаючи під’єднані пристрої, файли, папки, спільні мережеві ресурси, потужність процесора та кількісно визначені апаратні можливості. Це можливо. Незважаючи на це, програми, які працюють всередині контейнера, можуть отримати доступ лише до інформації про вміст контейнерів і призначені їм пристрої.

В операційних системах на базі Unix ця функція розглядається як розширена версія стандартного механізму chroot, який змінює кореневий каталог для запущеного процесу та його потомків. Крім механізмів ізоляції, ядро часто містить функції керування ресурсами, які обмежують вплив дій контейнера на інші контейнери. Віртуалізація, ізоляція та можливості керування ресурсами ядра є основою всіх контейнерів Linux.

Хоча контейнери зазвичай використовуються для опису систем віртуалізації на рівні ОС, вони також можуть посилатися на більш комплексні середовища віртуальних машин, які працюють у різному ступені узгодженості з головною ОС, наприклад, контейнери Microsoft Hyper V.

Незважаючи на відсутність повного доступу до ресурсів системи, комп’ютерна програма все ще може контролювати її в звичайних операційних системах персонального комп’ютера. Вони включають:

– адаптивні апаратні функції;

– дані, які можна інтерпретувати або кодувати;

– підключені пристрої, які дозволяють йому спілкуватися.

Залежно від програми та облікового запису користувача, операційна система може дозволити або обмежити доступ до певних ресурсів. Приховуючи ресурси, операційна система може гарантувати, що вони не будуть включені в результати перерахування комп'ютерної програми. З точки зору програмування, комп'ютерна програма використовувала ці ресурси, а операційна система контролювала взаємодію. Тим не менш, віртуалізація операційної системи або контейнеризація дозволяє запускати програми всередині контейнерів, яким виділено лише частину цих ресурсів. Запуск у контейнері дозволяє довільно розподіляти ресурси, що є єдиною інформацією, яку може бачити програма. Кожна операційна система має можливість створювати декілька контейнерів, кожному з яких виділяється частина ресурсів комп'ютера. У кожному контейнері можуть міститися різноманітні комп'ютерні програми. Вони можуть взаємодіяти одна з одною, паралельно або самостійно. Віртуалізація програми схожа на контейнеризацію. чому Одна комп'ютерна програма зберігається в ізольованому контейнері, при цьому файлова система є єдиною, яка зберігається окремо.

Середовище загального хостингу часто використовує віртуалізацію на рівні ОС для підвищення безпечного при розподілу ресурсів між великою кількістю користувачів, які не мають взаємної довіри. Системні адміністратори можуть використовувати його для об'єднання серверного обладнання шляхом перенесення служб на окремих хостах у контейнери на одному сервері.

Типові приклади включають розбиття кількох програм на окремі контейнери для підвищення безпеки, незалежності від апаратного забезпечення та функцій керування ресурсами. Однак переваги безпеки механізму chroot далеко не цілком залізні. Реалізації віртуалізації, здатні до живої міграції на рівні ОС, можна використовувати для динамічного балансування навантаження контейнерів між вузлами в кластері.

На відміну від повної віртуалізації, віртуальні розділи на рівні операційної системи не вимагають проміжної віртуальної машини чи емуляції, що призводить

до менших витрат для програм. Для роботи цього типу віртуалізації не потрібне апаратне забезпечення.

Інші методи віртуалізації є більш гнучкими, ніж віртуалізація на рівні операційної системи, оскільки вони не можуть розміщувати гостьові операційні системи, окрім хоста, або якщо вони мають спеціалізовані властивості, такі як основні ядра. На відміну від Windows, Linux може розмістити будь-який дистрибутив, але не всі. Віртуалізована архітектура створює обмеження для операційних систем, які працюють зі змінним введенням. Таким чином, ці програми використовують методи адаптації, включаючи аналітику ретрансляції хмарного сервера, для підтримки віртуального середовища на вищому рівні (ОС). Описані вище обмеження можна частково подолати за допомогою Solaris, який пропонує можливість створити контейнерне середовище, яке емулює старішу версію Solarus 8 або 9, якщо використовується цей хост SolarIS 10, використовуючи функцію фірмових зон. У системах Solaris, які використовують архітектуру x86, доступні фірмові зони Linux, які забезпечують повний простір користувача Linux і підтримку запуску програм Linux. Solaris також надає утиліти, необхідні для встановлення дистрибутивів Red Hat Enterprise Linux 3. x або CentOS. Кілька реалізацій пропонують механізми копіювання під час запису на рівні файлів. Простіше створювати резервні копії, споживати менше місця для зберігання та виконувати більше обслуговування кешу, ніж копіювання на рівні блоку. Тим не менш, віртуалізатори, які працюють з нерідними файловими системами, можуть генерувати та відкочувати знімки всього стану системи.

1.10 Емуляція

Комп’ютерну систему можна модифікувати, щоб вона функціонувала як інші комп’ютерні системи, за допомогою емулятора, який є різновидом апаратного чи програмного забезпечення в обчислювальній техніці. У більшості випадків емулятор надає хост-системі можливість запускати програми або периферійні пристрої, спеціально розроблені для гостьової системи. Процес імітації іншої

програми або пристрою комп’ютерною програмою в електронному пристрої називається емуляцією. Багато програм створено, щоб нагадувати принтери HP LaserJet, оскільки багато з них розроблено спеціально для виробників принтерів HP. Результати друку стороннього принтера HP можна відтворити за допомогою будь-якого програмного забезпечення, створеного для того самого пристрою HP.

Апаратний емулятор відноситься до пристрою, який працює як емулятор. Деякі комп’ютери Macintosh 1990-х років мали плату, сумісну з DOS, наприклад Centris 610 або Performa 630, що дозволяло їм запускати програмне забезпечення для ПК та апаратні емулятори на основі FPGA. На практиці це може бути складним завданням, особливо коли бажана поведінка системи не задокументована та потребує зворотного проектування. Без зазначення часових обмежень програмне забезпечення для емуляції може мати значні затримки, якщо воно не працює так швидко, як оригінальне обладнання. У тексті про це не йдеться. Емулятори зазвичай емулюють апаратну архітектуру, але якщо потрібне програмне забезпечення потребує мікропрограми або програмного забезпечення операційної системи, воно також має бути надано та може емулюватися. І ОС, і програмне забезпечення інтерпретуватимуться емулятором, а не запускатимуться на рідному обладнанні.

1.11 Віртуалізація серверів

Використання програмного забезпечення гіпервізора дозволяє відокремити різні віртуальні машини (сервери) від фактично довільних фізичних серверів, що відомо як віртуалізація серверів. Віртуальні сервери можуть керувати власними операційними системами.

Віртуалізація серверів дозволяє:

– Економія при придбанні серверів. Кожне завдання виконується віртуальним сервером з необхідними ресурсами (CPU, RAM тощо), а простої обладнання зменшуються;

– Зменшити навантаження на роботу інфраструктури. Загалом це простіше та швидше, ніж створення, видалення чи обслуговування віртуальної машини, порівняно з операціями, пов’язаними із запуском фактично ідентичних машин на місці;

– Підвищення відмовостійкості інфраструктури. Віртуальні машини не підключені одна до одної, тому, якщо одна з них зазнає програмного збою, це не вплине на роботу інших служб і програм.

Гіпервізор забезпечує кожну віртуальну машину ізольованим середовищем виконання, яке також обмежує доступ до апаратних ресурсів фізичного сервера як для віртуальних машин, так і для гостьових ОС. Гіпервізор дозволяє незалежну і паралельну роботу кількох операційних систем на одному комп’ютері. Відповідно до традиційного методу існує два типи гіпервізорів: ті, які працюють незалежно від апаратного забезпечення комп’ютера, і ті, для роботи яких потрібна операційна система хоста. Останніми роками класична класифікація зазнала змін із додаванням гібридного гіпервізора, який поєднує функції першого та другого типів.

Безпосередня робота з фізичним обладнанням головної машини можлива за допомогою «голого металевого гіпервізора», який також відомий як гіпервидимість типу 1. Для завантаження операційної системи потрібен гіпервізор типу 1. Найбільш ефективним і продуктивним є той, який використовує прямий доступ до вихідного обладнання без використання будь-якого зовнішнього програмного забезпечення. Гіпервізори, які встановлюються безпосередньо на фізичне обладнання, пропонують значні переваги безпеки. Оскільки кожна гостьова операційна система має власну віртуалізацію операційних систем (операційних систем або пристроїв), ризик атак, спрямованих на ці вразливості та недоліки безпеки, значно зменшується завдяки такому прийняттю. Гостьова віртуальна машина логічно ізольована від механізму віртуалізації хоста, що запобігає поширенню її атаки на інші машини на тому ж обладнанні. Для робочих навантажень виробничого рівня, які потребують більш тривалого часу безвідмовної роботи, покращених можливостей відновлення та інших важливих

функцій розробки від IT-організацій, гіпервізори типу 1 є кращим варіантом. Перший тип стандартного гіпервізора може розширюватися для віртуалізації робочих навантажень, які потребують кількох терабайтів оперативної пам’яті та сотень ядер ЦП. Крім того, гіпервізори типу 1 зазвичай підтримують програмно визначене сховище та мережу, що підвищує безпеку та портативність віртуалізованих робочих навантажень. Це особливо вигідно. Незважаючи на це, такі заходи вимагають значних початкових витрат. Гіпервізор типу 1 потребує додаткового зовнішнього керування, щоб максимізувати його можливості.

Гіпервізор типу 2 встановлюється на вже встановлену ОС. Через його залежність від ОС головної машини його іноді називають розміщеним гіпервізором, оскільки він керує викликами ЦП/Руху та Пам’яті (зберігання) і мережевих ресурсів. З початку віртуалізації x86 гіпервізор другого типу був реалізований як додатковий рівень поверх операційної системи існуючих систем. Незважаючи на те, що ціль і функція обох типів узгоджені, використання ОС із гіпервізорами типу 2 викликає неминучі переривання, оскільки кожен процес і елемент на кожній віртуальній машині повинні проходити через головну операційну систему. Крім того, якщо в операційній системі хоста є недоліки безпеки або помилки, усі віртуальні машини зверху можуть бути скомпрометовані. чому Гіпервізори типу 2 зазвичай зарезервовані для систем клієнтів або кінцевих користувачів, які пропонують кращу продуктивність і безпеку, а не використовуються в центрах обробки даних. Порівняно з гіпервізорами першого типу, вони дешевші та пропонують чудовий тестовий стенд. Віртуальні машини створюються розробниками програмного забезпечення для тестування програмного продукту перед випуском за допомогою «гіпервізора типу 2». виконує вправу з віртуальними машинами типу 2. Віртуальні робочі столи є можливим випадком їх використання. Гіпервізори типу 2 здатні працювати зі складними і великими кластерними середовищами.

Гібридний гіпервізор — це комбінація гіпервізор типу 1 і 2, яка працює на спеціалізованих сервісних або базових операційних системах. Це означає, що він використовує обидва типи можливостей для виконання різних завдань. У Hyper V

батьківський розділ або домен відомий як службова ОС із відповідним терміном dom0 у Xen. Ядро ОС активує режим підтримки віртуалізації після встановлення гіпервізора, який потім довіряє керування ресурсами процесора та пам’яті відповідній гіпервидимості. У той же час батьківський розділ бере на себе відповідальність за виклики драйверів пристроїв і операції вводу/виводу. Такий підхід вигідний для апаратної сумісності, оскільки немає необхідності додавати драйвери пристроїв до гіпервізора (список сумісного обладнання постійно розширюється). Сервісна ОС обробляє виклики драйверів пристроїв, що звільняє гіпервізор від обробки цих викликів.

Хоча віртуалізація з одним фізичним сервером є ефективною, вона має деякі очевидні недоліки. Наприклад, якщо сервер вийде з ладу під час встановлення, усі компоненти побудованої інфраструктури стануть непрацездатними. Кластер із кількох фізичних хостів — єдиний спосіб по-справжньому зрозуміти технологію віртуалізації. Група вузлів віртуалізації або віртуалізованих кластерів — це комбінація різних фізичних серверів, які представлені кінцевим користувачам як уніфіковані обчислювальні ресурси з однією точкою керування. Віртуальні машини запускаються на окремих фізичних серверах і можуть мігрувати між ними, що забезпечує високу доступність і гнучкий розподіл ресурсів.Майже всі перераховані нижще гіпервізори мають програмні механізми для об'єднання в кластер - VMWare vSphere, Hyper-V Failover Clustering Proxmox та ін.

Кластери віртуалізації переважно використовуються великими та середніми компаніями, які мають власну серверну інфраструктуру. Кластери віртуалізації також складають основу інфраструктури для хмарних провайдерів, які орендують сервери користувачам.

1.12 Віртуалізація пам'яті

Ресурси оперативної пам’яті в центрі обробки даних розділені на спеціалізовані віртуалізовані пули пам’яті, які може відкрити та розблокувати будь-який комп’ютер у кластері за допомогою віртуалізації пам’яті. До цього пулу

пам’яті звертається або сама операційна система, або програми, що працюють поверх ОС. Можна використовувати спільний пул пам’яті для таких завдань, як рівень обміну повідомленнями, високошвидкісний кеш або ресурс великого спільного сховища (GDMP) для додатків CPU або GPU.

Поширеним вузьким місцем у продуктивності програмного забезпечення є обмеження фізичної пам’яті, які можна подолати шляхом спільного використання пулу пам’яті між мережевими та розподіленими серверами за допомогою віртуалізації пам’яті. Ця інтегрована в мережу можливість дозволяє додаткам оптимізувати свою продуктивність, використовуючи великі обсяги пам’яті для покращення загальної продуктивності, продуктивності системи та ефективності системи, а також нових варіантів використання. Вузли у вузлах пулу зберігання (серверах) можуть використовувати програмне забезпечення для підключення до сховища, яке потім надає пам’ять, зберігає дані та отримує дані. Спільна пам’ять, вставка даних, видалення та політика надання для пристроїв зберігання даних (наприклад, файлів резервного копіювання), програмне забезпечення для керування пам’яттю та технології, які керують розподілом даних між вузлами, що вносять свій внесок, для надання кешу чи інших ресурсів, а також обробки запитів від процесорів клієнтських вузлів. . На рівні програми або операційної системи можна створити пул пам’яті. За допомогою API або мережевої файлової системи пул використовується на рівні програми для встановлення масштабованості спільного кеша пам’яті. Це робиться через Кеш сторінок може використовувати пул для зберігання неймовірно швидких ресурсів пам’яті, які є набагато швидшими, ніж локальне чи мережеве сховище, на рівні операційної системи.

Реалізації віртуалізації пам'яті відрізняються від систем спільної пам'яті. Системи спільної пам’яті не можуть абстрагувати ресурси пам’яті, тому вони повинні бути реалізовані в одному екземплярі операційної системи, а не в середовищах згрупованих програм. Це проблематично.

Залежно від способу структуризації віртуального адресного простору, що визначає перетворення віртуальних адрес в фізичні, виділяється три класи віртуальної пам'яті:

– Сторінкове розподіл. Одиницею переміщення між пам'яттю і диском є сторінка - частина віртуального адресного простору фіксованого і невеликого обсягу.

– Сегментний розподіл. Одиницею переміщення між пам'яттю і диском є сегмент - частина віртуального адресного простору довільного обсягу, що містить осмислену з деякою точки зору сукупність даних (підпрограму, масив і т.д.);

– Сегментно-сторінкове розподіл. Об'єднує елементи попередніх класів. ВАП структурується ієрархічно: ділиться на сегменти, а потім сегменти поділяються на сторінок. Одиницею переміщення між пам'яттю і диском є сторінка.

1.13 Віртуалізація в бізнесі

Серверна інфраструктура підприємства зазвичай віртуалізована більшістю компаній, коли йдеться про віртуалізацію. Незважаючи на це, існує безліч рішень, які використовують віртуальні машини для кінцевих користувачів і можуть значно підвищити продуктивність робочої сили організації. Яким чином віртуальні машини використовуються в бізнес-додатках на настільних платформах для користувачів:

– Розробка архівних копій типових шаблонів для робочих середовищ користувачів. Специфіка роботи організації вимагає від співробітників використання певного набору програмного забезпечення. Після того, як новий співробітник приєднається до організації, він повинен встановити операційну систему та налаштувати її відповідно до вимог компанії та політики безпеки, а також встановити все необхідне прикладне програмне забезпечення. Прямим вирішенням цієї проблеми є використання шаблонів віртуальних машин шляхом встановлення платформи віртуалізації робочого столу в реальному часі для співробітника та створення спеціалізованих віртуальних машин із різних організаційних шаблонів. Цей процес включає встановлення програмного забезпечення та налаштування операційних систем. Ця модель може значно скоротити час розгортання, а також забезпечує значну гнучкість під час

перенесення віртуального робочого столу співробітника на іншу фізичну машину. Використання цього сценарію може вимагати більшої кількості апаратних ресурсів на обладнанні, але це більш вигідно, коли співробітникам доводиться керувати величезними обсягами різноманітних даних, що вимагає часу для резервного копіювання. Співробітники відділу маркетингу, які встановлюють різні програми та працюють з різними видами документів, щодня щось пробують. Вони можуть скопіювати папку з файлами своєї віртуальної машини наприкінці робочого дня, не турбуючись про можливі збої під час встановлення програми завтра;

– Розробка системи віртуального робочого столу, яка дозволяє безпечно зберігати налаштовані середовища на корпоративних серверах. Кінцеві користувачі використовують віддалений доступ до навколишнього робочого столу, який досягається за допомогою служб терміналів у корпоративному центрі обробки даних. Використання віртуальних машин є дорогим вибором, оскільки потребує підтримки серверних платформ віртуалізації корпоративного центру обробки даних. Однак у цьому випадку можна забезпечити найкращу безпеку та конфіденційність. Імовірність витоку конфіденційної інформації значно зменшується, оскільки всі робочі середовища централізовано зберігаються та обслуговуються в центрі обробки даних, який захищено заходами безпеки. Цей рівень доступності підвищує значною мірою завдяки тому факту, що доступ може бути гарантований з будь-якого місця з високошвидкісним Інтернетом. Це рішення не підходить для всіх платформ віртуалізації робочого столу, але воно актуальне для кінцевих користувачів. Рішенням цієї проблеми є VMware Virtual Desktop Infrastructure що базується на віртуальній інфраструктурі серверів у корпоративному ЦОД. Структура такої моделі зображено на рис. 1.1;

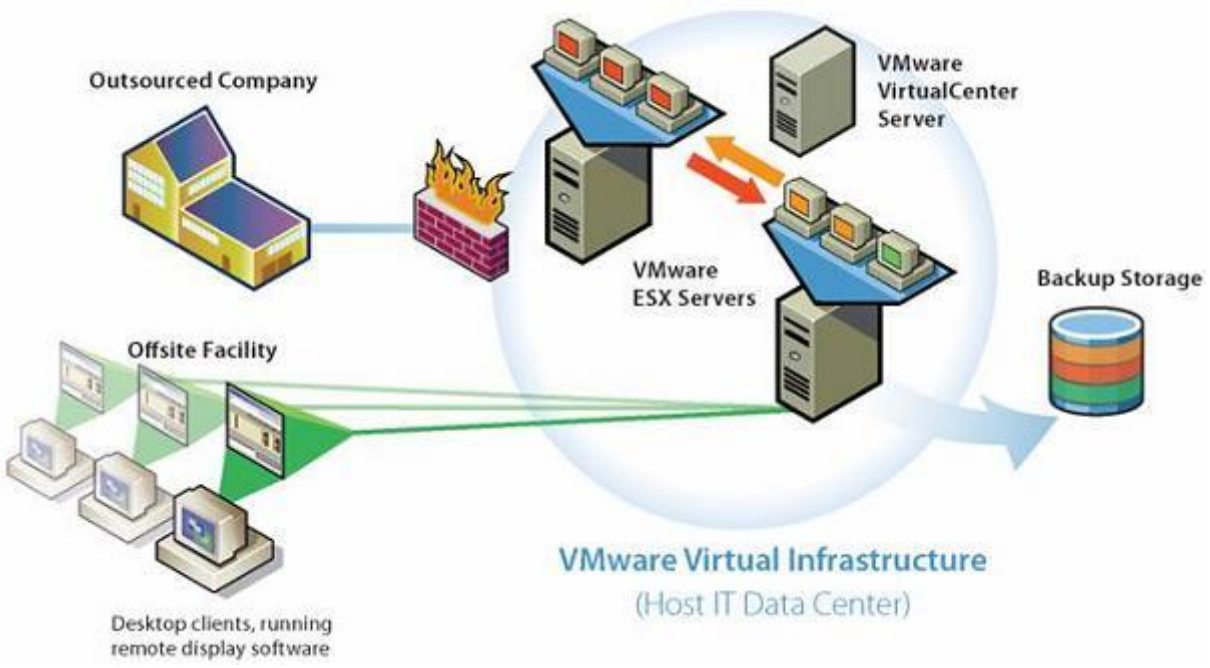


Рисунок 1.1– Структура віртуальної інфраструктури десктопів

– Захист віртуальних машин через використання. Якщо від IT-фахівців вимагається зберігати конфіденційність особистої інформації та демонструвати клієнтам різноманітні опції програмного забезпечення, вони добре підходять для створення захищених віртуальних машин, які надають спосіб розмежувати доступ до різних функцій під час роботи з віртуальними машинами. Варіантом є створення періоду дії віртуальної машини та розповсюдження його довільним користувачам. Прикладом такого рішення є продукт ACE Manager компанії VMware;

– Передача можливостей навчання для користувачів через віртуальні машини з ОС і встановленим прикладним програмним забезпеченням для полегшення їхнього розвитку. Організація може встановити платформу віртуалізації на всіх комп'ютерах у класі, і можна створити шаблон для навчання своїх співробітників використанню будь-якого програмного забезпечення чи продукту. Потім віртуальну машину можна скопіювати на кожен комп'ютер і запустити з усіма необхідними програмами. Створіть новий шаблон віртуальної машини та поширте його на всі комп'ютери в класі, якщо вам потрібно навчити інший продукт. Розробка та тестування програмного забезпечення всередині організації. На відміну від інших віртуальних машин, вони пропонують ідеальне середовище для розробки програмного забезпечення. Поведінку програми на різних платформах

можна емулювати шляхом впровадження відповідних операційних систем та інтерфейсів користувача за потреби. Це дозволяє проводити гнучке тестування на кількох пристроях. Цей варіант використання дозволяє моделювати робочі з'єднання віртуальних машин на фізичній платформі, яка взаємодіє з мережами одна одної. Крім того, деякі платформи, наприклад VMware Workstation, пропонують можливість створювати дерева стану віртуальних машин, які містять певну конфігурацію, визначену користувачем. За допомогою одного клацання миші ви можете повернутися до всіх цих станів. Приклад такого дерева зображено на рис. 1.2

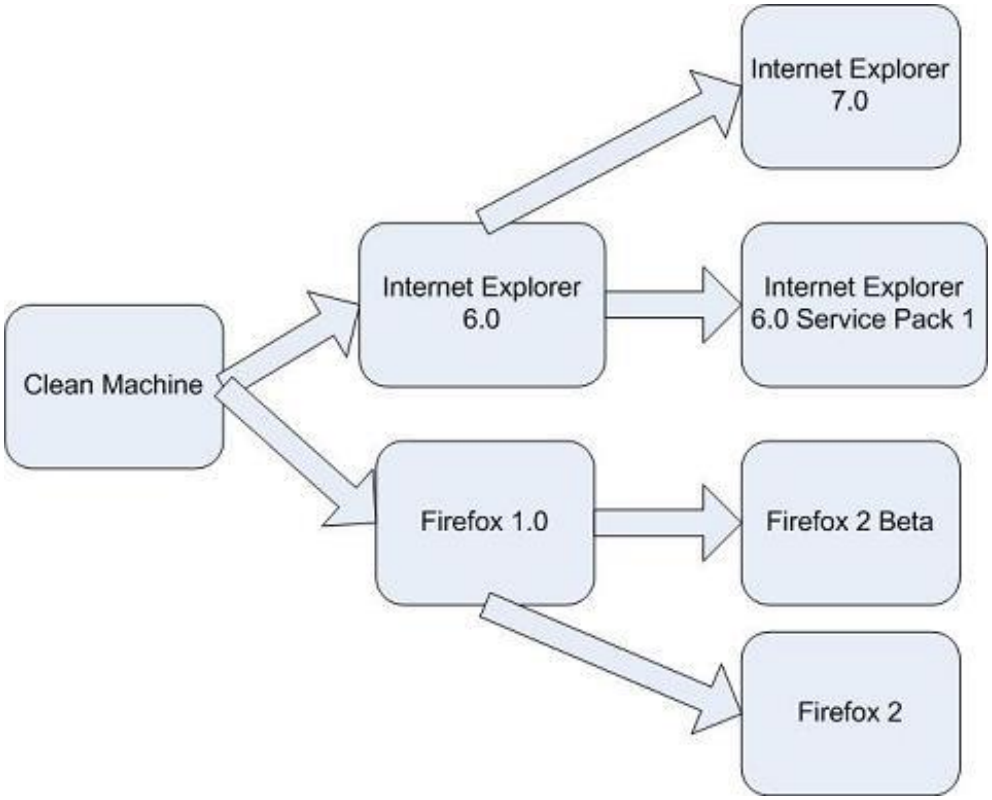


Рисунок 1.2– Дерево станів віртуальних машин

РОЗДІЛ 2 РЕАЛІЗАЦІЯ СУЧАСНИХ ТЕХНОЛОГІЙ ВІРТУАЛІЗІЦІЇ

2.1 Віртуалізація за допомогою Oracle VM VirtualBox

VirtualBox — це потужний продукт віртуалізації x86 і AMD64/Intel64 для корпоративного та домашнього використання. Гіпервізор типу 2, який також відомий як розміщений гіпервізор, є Oracle VM VirtualBox. У той час як гіпервізор «голий метал» або гіпервізор типу 1 буде працювати безпосередньо на апаратному забезпеченні Oracle VM VirtualBox потребує встановлення вже встановленої ОС. Хост може забезпечити запуск існуючих програм. Усі хост-платформи підтримують Oracle VM VirtualBox, і використовуються однакові формати файлів і зображень. Використовуючи іншу хост-ОС, віртуальні машини, створені на одному хості, можуть працювати на іншому хості. Як приклад, ви можете створити віртуальну машину в Windows, а потім працювати з нею під Linux. Крім того, віртуальні машини можна легко імпортувати та експортувати у відкритому форматі.

Спільні папки, безшовні вікна та 3D-віртуалізація як гостьові програми. Oracle VM VirtualBox Guest Additions — це пакети програмного забезпечення, які можна інсталювати всередині підтримуваних гостьових систем, щоб покращити їхню продуктивність і забезпечити додаткову інтеграцію та зв’язок із хост-системою. Після встановлення гостьових доповнень віртуальна машина матиме можливість автоматично регулювати роздільну здатність відео разом із безшовними вікнами та прискореною 3D-графікою. Гостьова машина може отримати доступ до файлів на хост-системі зі спільними папками, наданими гостьовими програмами.

Чудова апаратна підтримка. Серед інших можливостей Oracle VM VirtualBox підтримує наступне.:

– Гостьова багатопроцесорна (SMP). Незалежно від того, скільки ядер ЦП є на вашому хості, Oracle VM VirtualBox може забезпечити 32 віртуальних ЦП на віртуальну машину;

– Підтримка USB-пристроїв. Використання віртуального USB-контролера забезпечує Oracle VirtualBox, у якому пристрій можна підключити до віртуальних машин за допомогою зовнішнього USB-порту, без необхідності встановлення драйверів для пристрою на хості. Деякі типи пристроїв не підтримують USB;

– Апаратна сумісність. VirtualBox для Oracle VM забезпечує підтримку віртуалізації для широкого діапазону пристроїв, у тому числі багатьох, які зазвичай пропонують конкуруючі віртуалізатори. Серед включених функцій різноманітні віртуальні мережеві та звукові карти, віртуальні послідовні/паралельні порти (VNR) і розширений програмований контролер переривань вводу/виводу, який є в багатьох комп'ютерних системах. Oracle VM VirtualBox може імпортувати віртуальні машини сторонніх розробників, оскільки образи дисків із реальних пристроїв легко копіюються;

– Повна підтримка ACPI. Інтерфейс розширеної конфігурації та живлення (ACPI) повністю підтримується Oracle VM VirtualBox. Oracle VM VirtualBox полегшує клонування образів дисків із реальних або сторонніх віртуальних машин. Oracle VM VirtualBox має ексклюзивну підтримку статусу живлення ACPI, що дозволяє гостьовим ОС отримувати сповіщення про споживання енергії хостом. Мобільні системи з живленням від акумулятора дозволяють гостям увімкнути енергозбереження та повідомляти користувача про будь-який залишок заряду, особливо в повноекранних режимах;

– Багатоекранна роздільна здатність. Завдяки роздільній здатності екрану, яка у багато разів перевищує роздільну здатність фізичного екрана, віртуальні машини Oracle VM VirtualBox можна спільно використовувати між численними екранами, підключеними до систем емуляції на різних хост-системах;

– Вбудована підтримка iSCSI. Використовуючи цю функцію, віртуальну машину можна підключити безпосередньо до внутрішнього сервера зберігання iSCSI із хост-системи. Це дозволяє віртуальній машині отримувати прямий доступ до цілі iSCSI без додаткових витрат на віртуалізацію жорстких дисків у файлах-контейнерах;

– Завантаження мережі PXE. NIC Oracle VM VirtualBox цілком можуть завантажуватися віддалено за допомогою середовища виконання перед завантаженням (PXE);

– Розгалужені знімки кількох поколінь. Oracle VM VirtualBox має можливість зберігати випадкові знімки стану віртуальних машин. Можна створити ціле дерево знімків, відновивши віртуальну машину до того знімка, який ви раніше зберегли, а потім запустивши іншу конфігурацію MVVM із цього знімка. Робота віртуальної машини дозволяє створювати та видаляти знімки;

– Групи віртуальної машини. Користувач може керувати та групувати віртуальні машини в Oracle VM VirtualBox за допомогою функції групування, яка також дозволяє індивідуальне керування. Будь-яка віртуальна машина може входити до кількох груп, і групи структуровані як ієрархія. Ви можете мати безліч груп. Групи зазвичай здатні виконувати завдання, ідентичні завданням окремих віртуальних машин, включаючи запуск, призупинення, перезапуск, виконання, скасування збереженого стану, відображення у файловій системі, сортування;

– Чиста архітектура та безпрецедентна модульність. Oracle VM VirtualBox має високу модульність, має чіткі внутрішні інтерфейси програмування та чітке розмежування між кодом клієнта та сервера;

– Завдяки модульній архітектурі Oracle VM VirtualBox можна інтегрувати з іншими програмними системами за допомогою потужного програмного забезпечення (SDK), забезпечуючи повну функціональність і налаштування;

– Віддалений дисплей машини. Розширення віддаленого робочого стола VirtualBox (VRDE) забезпечує віддалений доступ до будь-якої віртуальної машини, яка зараз активна, і забезпечує високопродуктивні можливості. Це розширення розроблено для повної підтримки USB-клієнта разом із підтримкою оригінального протоколу віддаленого робочого стола (RDP) Microsoft Windows;

– Використання сервера RDP, вбудованого в Microsoft Windows, не присутнє у VRDE. Рівень віртуалізації не пов’язаний безпосередньо з VRDE. Він може працювати в гостьових ОС, відмінних від Windows, у тому числі в текстовому режимі, і не потребує підтримки програм у віртуальній машині.

2.1.1 Створення віртуальної машини

Для створення віртуальної машини потрібно дотримуватись послідовності.

Крок 1. У менеджері VirtualBox натисніть «Створити». Буде присутній майстер, який допоможе вам налаштувати нову віртуальну машину, зображено на рис. 2.1.

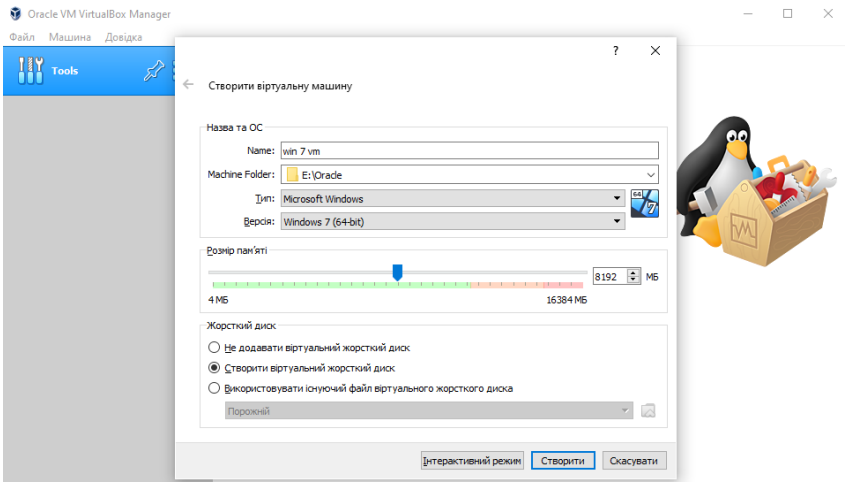


Рисунок 2.1– Створення віртуальної машини

Крок 2. Дайте ім’я вашій віртуальній машини. Це ім’я використовується для ідентифікації віртуальних машин і дискових файлів, і воно відображається у списку машин, доступних у вікні VirtualBox Manager.

Крок 3. Визначте розташування віртуальних машин на вашому комп’ютері.

Крок 4. Визначте тип операційної системи яку буде використовувати ваш віртуальний ком’ютер.

Крок 5. Укажіть обсяг оперативної пам’яті, який Oracle VM VirtualBox має зберігати під час завантаження. Ваш хост-комп’ютер забере пам’ять, виділену вам у цьому прикладі, і ці дані відображатимуться в гостьовій операційній системі як оперативна пам’ять, встановлена на віртуальній машині. Переконайтеся, що хост-ОС має достатньо оперативної пам’яті. Недостатня кількість оперативної пам’яті може призвести до того, що система перезапише пам’ять на жорсткий диск, що призведе до зупинки роботи головної системи. Ви можете змінити цей параметр, а також інші параметри після встановлення віртуальної машини.

Крок 6. Натиснути на вікно «Створити».

Крок 7. Вибрати обсяг диску для віртуальної машини.

Крок 8. Потім вам потрібно вказати віртуальний жорсткий диск, на якому буде побудована ваша віртуальна машина, зображено на рис. 2.2. Oracle VM VirtualBox підтримує такі типи файлів:

– Файл, який динамічно призначається, лише збільшується в розмірі, коли гостьова система знаходиться на своєму віртуальному жорсткому диску для зберігання даних. Коли диск заповнюється даними, це призводить до збільшення розміру файлу.

– Файл фіксованого розміру. Під час використання віртуального жорсткого диска вказаний файл негайно завантажується, використовуючи фактично лише десять відсотків. Файли фіксованого розміру займають набагато більше місця, але накладні витрати, пов’язані з динамічно розподіленими файлами, значно нижчі, що призводить до трохи вищої продуктивності.

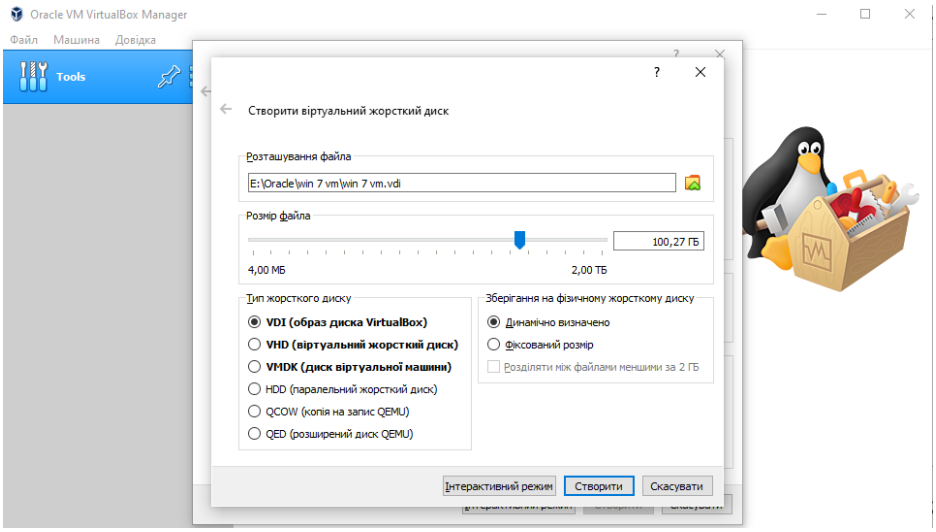


Рисунок 2.2– Створення віртуального жорсткого диску

Крок 9. Натиснути вікно «Створити» для завершення початкових налаштувань та створення віртуальної машини, зображено на рис. 2.3.

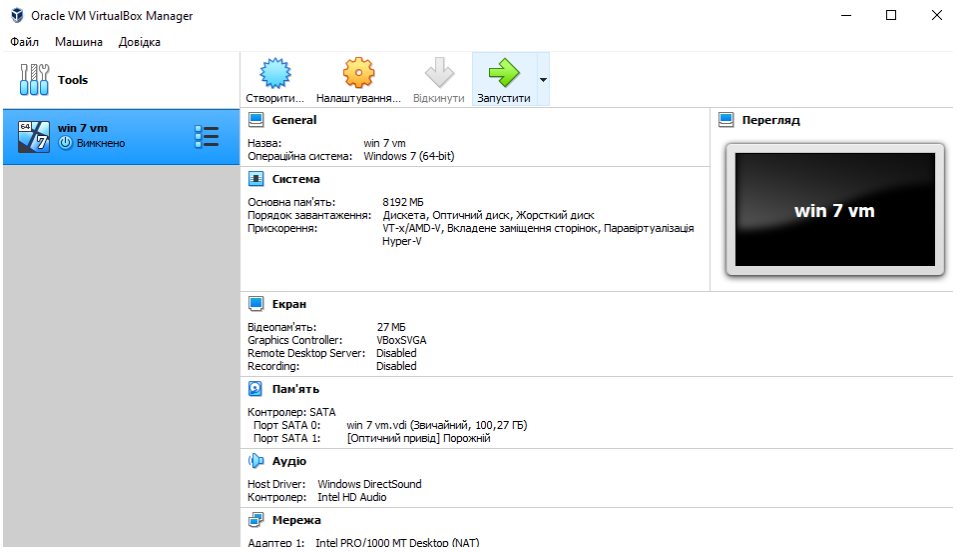


Рисунок 2.3– Створена віртуальна машина

2.1.2 Встановлення операційної системи на віртуальну машину

Інсталяція ОС на віртуальній машині не відрізняється суттєво від звичайної інсталяції ОС.

Крок 1. На початку нам потрібно натиснути «Запустити», щоб запустити віртуальну машину..

Крок 2. Необхідно вибрати образ бажаної операційної системи в нашому випадку це Windows 7, зображено на рис. 2.4.

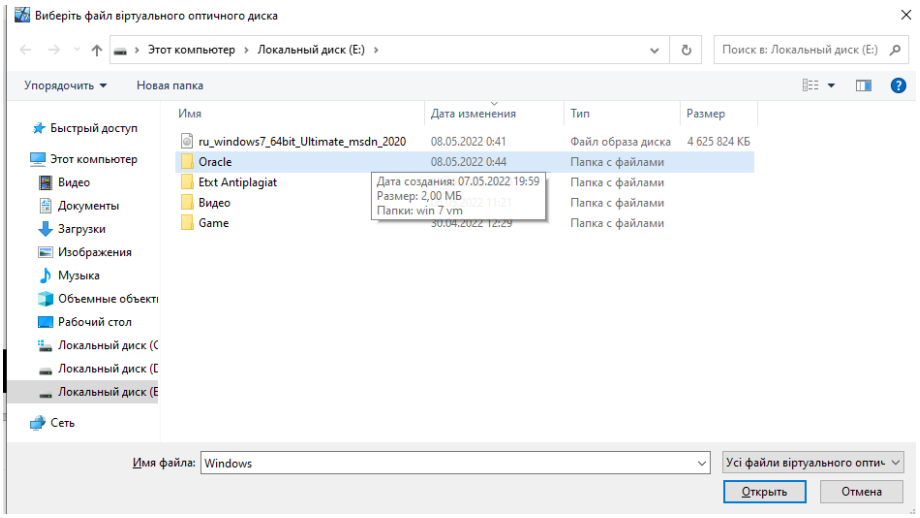


Рисунок 2.4 – Образ ОС

Крок 3. Починаємо встановлення операційної системи натиснувши кнопку «Запустити».

Крок 4. Встановлюємо операційну систему, зображено на рис. 2.5.

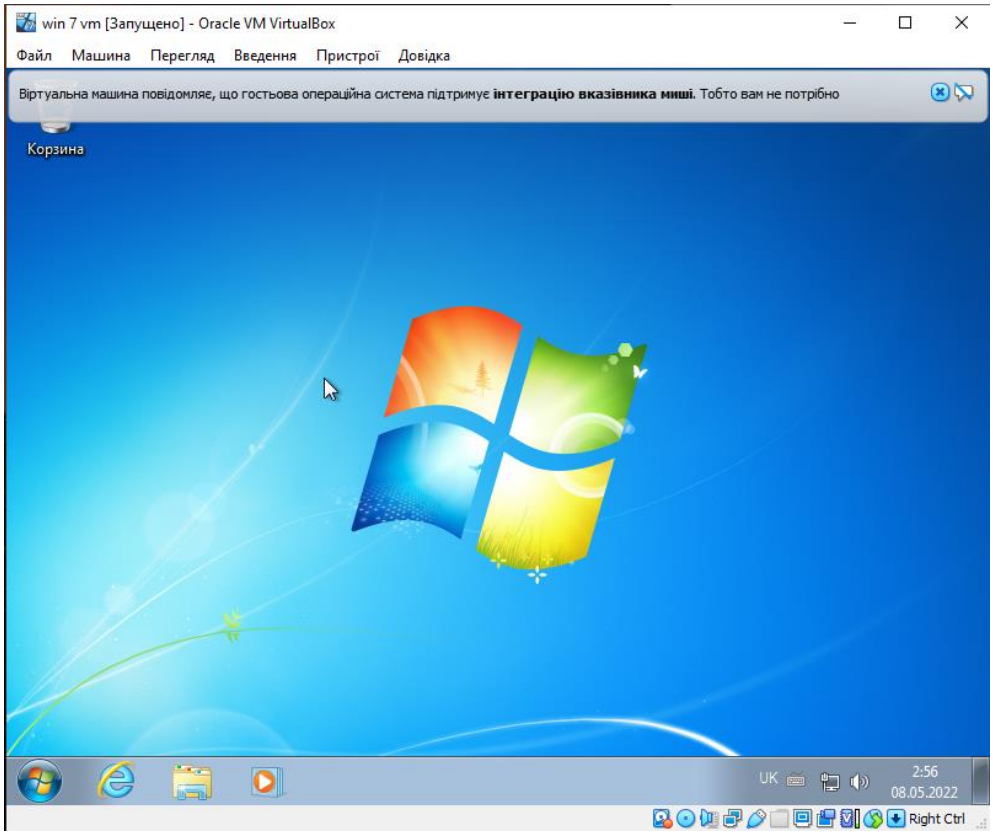


Рисунок 2.5 – Встановлена ОС на віртуальну машину

2.1.3 Встановлення VirtualBox Extension Pack

VirtualBox Extension Pack — Ці розширення базуються на пакеті Oracle VM VirtualBox, додаючи додаткові функції. Додаткові функції включені в пакет розширення:

– Віртуальний пристрій USB 2.0 (EHCI);

– Віртуальний пристрій USB 3.0 (xHCI);

– Підтримка протоколу віддаленого робочого столу VirtualBox (VRDP);

– Перехід через веб-камеру хоста;

– Завантажувальна ПЗУ Intel PXE;

– Шифрування образу диска за допомогою алгоритму AES;

– Функції хмарної інтеграції.

Крок 1. Зайти в налаштування Oracle VM VirtualBox Manager.

Крок 2. Зайти в «Розширення» та додати VirtualBox Extension Pack .

2.1.4 Налаштування Мережі та RDP

Протокол віддаленого робочого столу (RDP) — Microsoft розробила власний протокол, який пропонує графічні інтерфейси для підключення користувачів до іншого комп’ютера через мережеве з’єднання. Для цього користувач використовує клієнтське програмне забезпечення RDP, а серверне програмне забезпечення RDH має бути запущено на іншому комп’ютері. Отже, Microsoft Windows, Linux, Unix та різні інші операційні системи мають клієнтів для більшості з них iOS, Android тощо.

Крок 1. Виберіть віртуальну машину в Oracle VirtualBox Manager і перейдіть до налаштувань.

Крок 2. В підрозділі "Мережа" виберіть "Лише головний адаптер" та "Virtualbox Host-Only Ethernet Adapter", зображено на рис. 2.6.

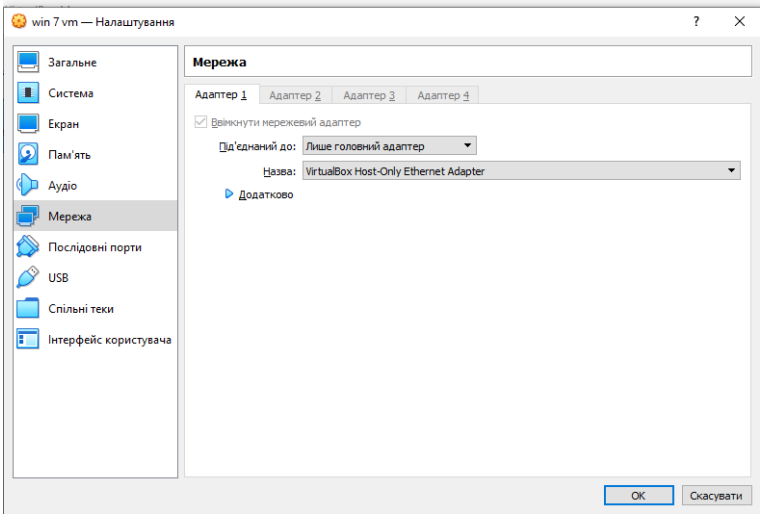


Рисунок 2.6 – Налаштування адаптера мережі віртуальної машини

Крок 3. На хостовій машині треба зайти в "Панель Управління" потім в "Мережа та інтернет" та "Мережеві підключення" обрати адаптер хостової машини та у властивостях у підрозділі "Доступ" поставити галочку біля "Дозволити іншим користувачам мережі використовувати підключення до Інтернету даного комп'ютера" та натиснути "Ок", зображено на рис. 2.7.

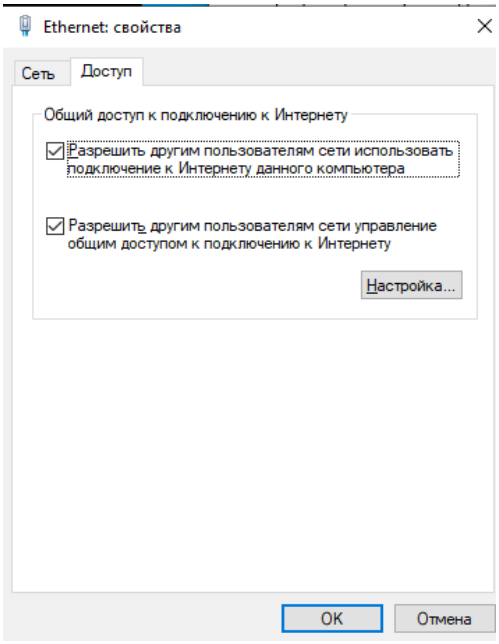


Рисунок 2.7 – Налаштування адаптера мережі хоста

Крок 4. Перейдіть до властивостей мережевого адаптера на цій віртуальній машині та укажіть IP-адресу адаптера , зображено на рис. 2.8.

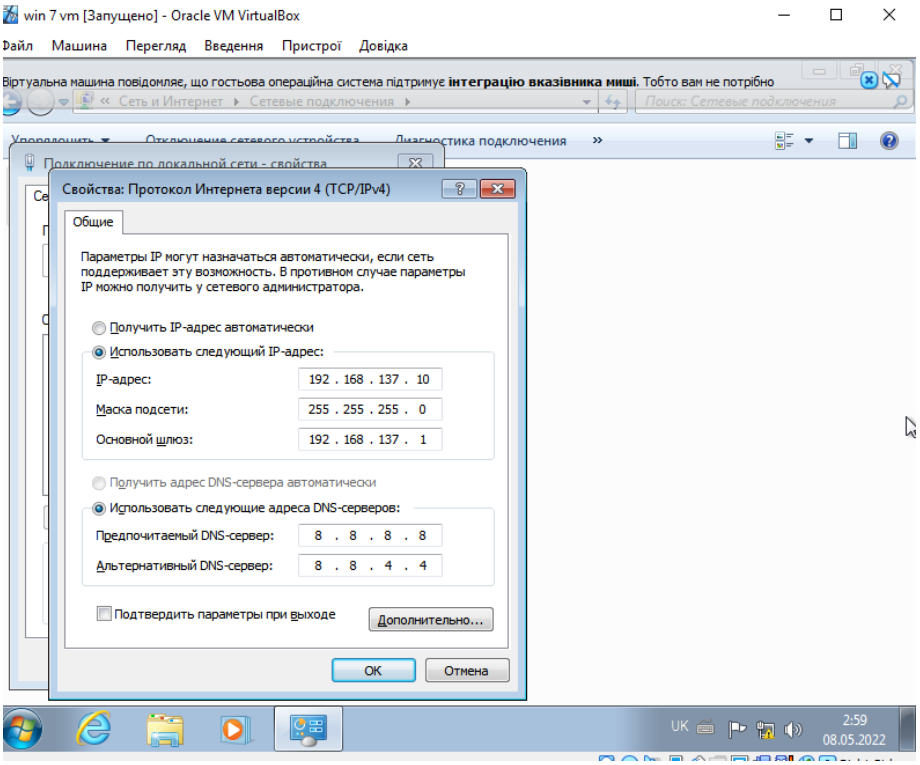


Рисунок 2.8 – Налаштування IP-адресу віртуальної машини

Крок 5. Заходимо в властивості "Мій комп'ютер" потім в "Налаштування віддаленого доступу" та вибираємо "Дозволити підключення від комп'ютерів з будь-якою версією віддаленого робочого столу" та натискаємо "Ок", зображено на рис. 2.9.

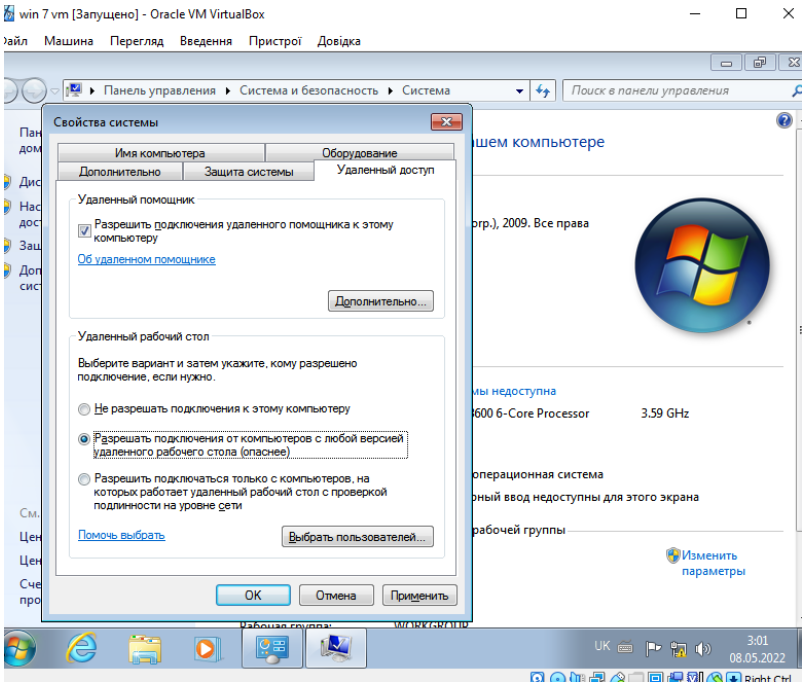


Рисунок 2.9 – Налаштування віддаленого доступу

Крок 6. Перейдіть до адміністрування комп'ютера та виберіть «Користувачі». Створіть нового користувача з призначеним паролем, зображено на рис. 2.10.

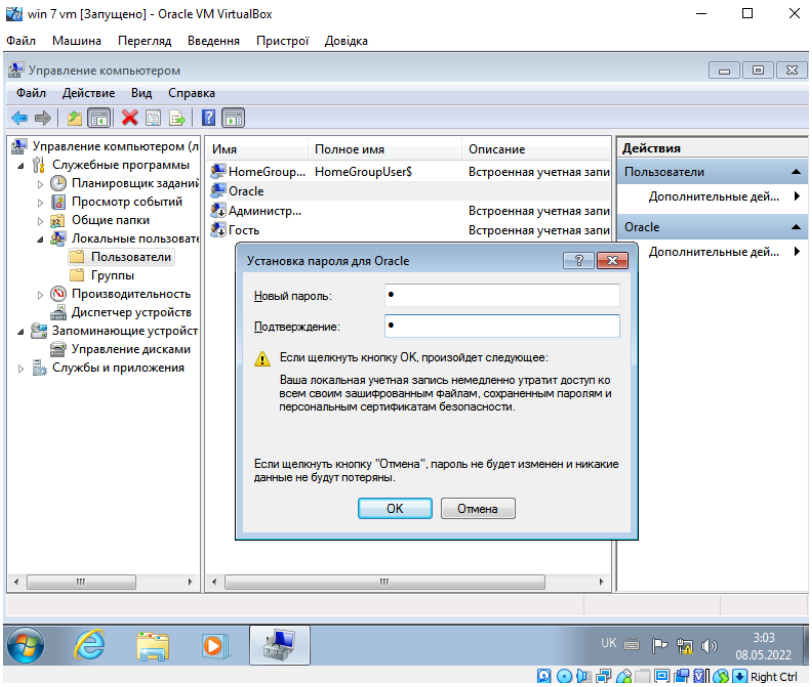


Рисунок 2.10 – Створення нового користувача

Крок 7. Робимо перевірку. За допомогою клавіш WIN+R відкриваємо утиліту "Виконати", в рядку вводимо команду "mstsc". Вводимо IP-адресу нашої віртуальної машини а саме 192.168.137.10 та вводимо ім'я та пароль користувача якого ми створили, зображено на рис. 2.11.

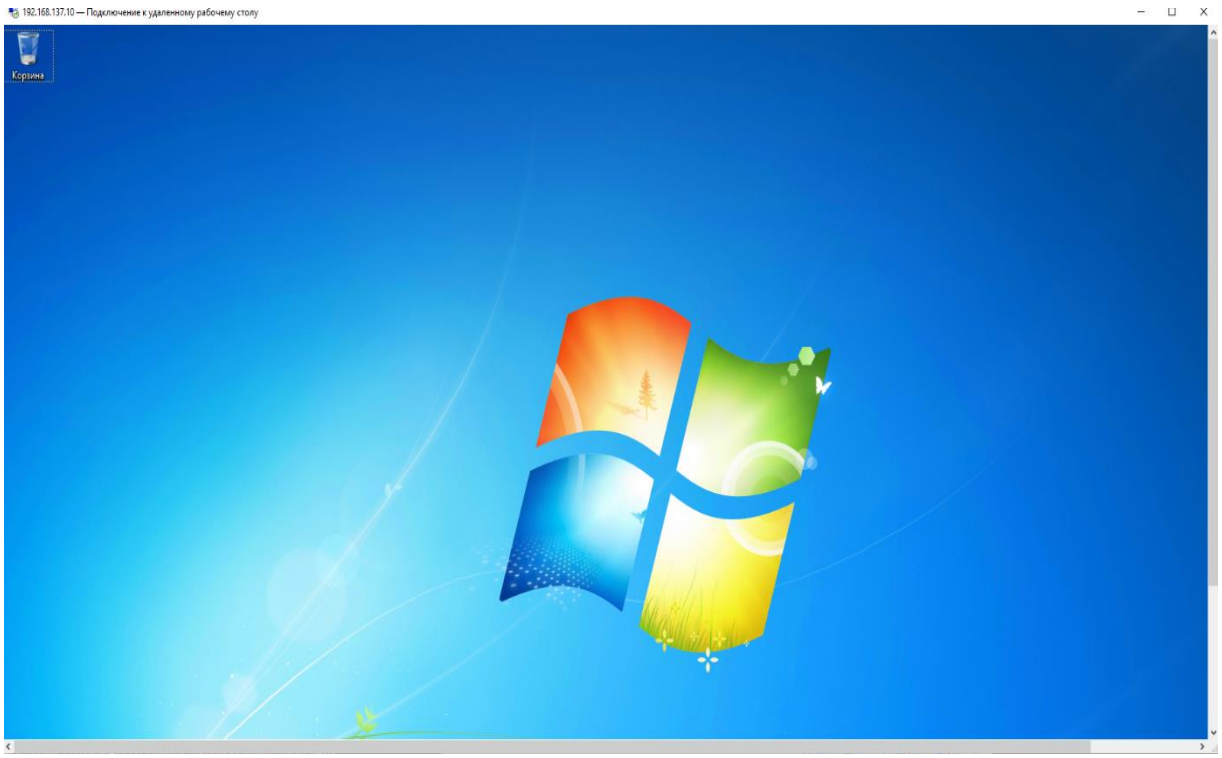


Рисунок 2.11 – Віддалене підключення до віртуальної машини

2.1.5 Налаштування віртуальної машини

Гостьова ОС може отримувати події миші з віртуального планшета USB, наданого Oracle VM VirtualBox, який використовується для нових віртуальних машин. Наведений нижче приклад стосується пристроїв, здатних підтримувати підтримку миші з коробки в сучасній гостьовій ОС без необхідності захоплення миші.

Якщо віртуальна машина розпізнає лише стандартні миші та клавіатури PS/2, ОС візьме на себе виключний контроль над вашою клавіатурою та мишею, оскільки вона не знає, що вони працюють не на реальному комп’ютері. Інші програми та потенційно інші віртуальні машини на вашому хості повинні спільно використовувати клавіатуру та мишу з вашою віртуальною машиною, якщо вона не працює в повноекранному режимі.

Ваша віртуальна машина або апаратне забезпечення іншого комп’ютера відповідає за керування клавіатурою та мишею, за умови, що воно було встановлено разом із гостьовими доповненнями та після встановлення ОС.

Клавіатура та миша не є взаємовиключними. Вікно віртуальної машини завжди обмежене другим вказівником миші, який ви можете переглядати. Віртуальна машина активується клацанням по ній.

Oracle VM VirtualBox зарезервував ключ "host" на вашій клавіатурі, щоб повернути клавіші клавіатури та миші до вашої хост-ОС. Ця клавіша Ctrl автоматично встановлюється як правильна на вашій клавіатурі. Ліва клавіша Command є типовою клавішею хоста на Mac. Цей параметр можна змінити в глобальних налаштуваннях Oracle VM VirtualBox. Поточні налаштування ключа хоста завжди видно в нижньому правому куті..

Певні процедури вимагають активації певних комбінацій клавіш у певних операційних системах. Залежно від операційної системи віртуальна машина може відображати комбінації клавіш, які можна використовувати для завантаження. Комбінація клавіш є одним із факторів, які визначають одержувача цих натискань.

Кілька комбінацій клавіш зарезервовано для головних операційних систем. Комбінація клавіш для перезавантаження гостьової ОС у вашій віртуальній машині зазвичай прикріплена до головної ОС, і Ctrl+Alt+Delete не підтримується. Незважаючи на те, що Windows і Linux перехоплюють цю важливу комбінацію, головна ОС буде лише перезавантажена.

Щоб перезапустити систему X Window на хостах Linux і Oracle Solaris, звичайною процедурою є використання Ctrl+Alt+Backspace, щоб скинути сервер і відновити графічний інтерфейс. Оскільки X-сервер отримує цю комбінацію, її натискання та утримання зазвичай перезапускає графічний інтерфейс хоста та вимикає всі запущені програми, включаючи Oracle VM VirtualBox..

Доступ до віртуальних терміналів на хостах Linux зазвичай здійснюється за допомогою Ctrl+Alt+Fx, яка є однією з функціональних клавіш для перемикання між типами віртуальних терміналів. Fx включено в цю комбінацію. Головна ОС завжди перемикатиме головні термінали після розпізнавання натискань клавіш, як і Ctrl+Alt+Delete. Ось чому він працює інакше. Ці натискання клавіш можна надіслати до гостьової ОС у віртуальній машині одним із наведених нижче методів.

Для керування вікном віртуальної машини використовуйте пункти меню «Введення» та «Клавіатура». У цьому меню ви знайдете параметри «Вставити» Ctrl+Alt+Delete та «Backspace» як «Застосувати» для видалення. Останній вибір обмежено гостьовими системами Linux або Oracle Solaris. У це меню можна включити ярлик Хост.

Ви також можете змінити знімний носій у меню «Пристрої» у вікні віртуальної машини, коли віртуальна машина запускається. Параметри для вибору в деталях, що Oracle VM VirtualBox буде виводити на вашу віртуальну машину як CD, DVD або дисковод. Параметри віртуальної машини такі ж, як і в діалоговому вікні «Параметри» основного вікна Oracle VM VirtualBox. Коли віртуальна машина перебуває в стані «Запущено» або «Збережено», вона вимикає діалогове вікно налаштувань, щоб ви могли використовувати її меню «Пристрої» замість того, щоб вимагати вимикати та перезавантажувати операційну систему кожного разу, коли ви хочете змінити носій.

Розмір вікна віртуальної машини можна змінювати під час її роботи. Якщо все зроблено правильно, вікно буде масштабуватися таким чином:

– Екран віртуальної машини буде збільшено до розміру вікна, коли активовано режим масштабування. Якщо у вас працює кілька машин і ви хочете переглядати одну у фоновому режимі, ця функція може бути корисною. Якщо вихідний екран віртуальної машини дуже малий, можливо, буде корисно максимізувати розмір вікна під час використання старішої операційної системи;

– Додатки, які підтримують автоматичне змінення розміру, можуть автоматично регулювати роздільну здатність екрана гостьової ОС, якщо вони встановлені. Якщо налаштувати розмір вікна віртуальної машини до 100 пікселів під час використання гостьового режиму Windows із роздільною здатністю 1024x768 пікселів, роздільна здатність дисплея Windows буде змінена на 1124x768 відповідно до гостьових програм;

– Якщо екран віртуальної машини не є більшим за нього, необхідно буде відцентрувати екран. До вікна машини буде додано менші смуги прокрутки.

2.2 Встановлення VMware ESXI за допомогою VMware Workstation

VMware Workstation Pro (відомий як VMware Workstation) — це розміщений гіпервізор, який підтримує x64-версії операційних систем Windows і Linux, користувачі можуть створювати віртуальні машини на власній фізичній машині та використовувати їх разом із хост-машиною. Кожна віртуальна машина має можливість запускати різні версії Microsoft Windows і Linux, а також BSD і MS DOS.

Використання VMware Workstation дає змогу спільно використовувати фізичні диски та USB-пристрої з віртуальною машиною за допомогою існуючих мережевих адаптерів. Файл можна використовувати як віртуальний оптичний привід, файл ISO-образу сумісний із ним, а віртуальні жорсткі диски реалізовано як файли default. vmdk. У будь-який момент часу VMware Workstation Pro може зберегти стан віртуальної машини. Відновлення цих знімків гарантує, що віртуальна машина повернеться до свого попереднього стану без змін після того, як її зробили віртуальні машини.

Папку інвентаризації у VMware Workstation можна використовувати для групування кількох віртуальних машин. Можливість вмикати та вимикати комп’ютери в одному об’єкті в межах «папки» особливо корисна для тестування складних середовищ клієнт-сервер..

У платформі віртуалізації VMware vSphere є відкритий гіпервізор. На відміну від аналогів віртуальних машин, VMware ESXi — це звичайний гіпервізор, який можна створити та запускати без операційної системи, але натомість працює поверх апаратного забезпечення. Цей прямий зв’язок із апаратним забезпеченням дозволяє йому працювати ефективніше, працювати на вищих швидкостях і бути більш гнучким, ніж інші гіпервізори». У поєднанні з іншими компонентами платформи VMware vSphere дуже вигідно використовувати ESXi від VMware для масштабних інфраструктур віртуальних робочих столів.

VMkernel, який є мікрокінелем, який запускає VMware ESXi, відповідає за керування обладнанням, гостьовими системами та сервісним рівнем платформи

віртуалізації VMware. VMkernel керує фізичним обладнанням і гостьовими віртуальними машинами, встановлює прямі з'єднання з процесором і пам'яттю та покладається на модулі, пов'язані через інший модуль для доступу до обладнання.

Примітні особливості VMware ESXi включають:

– Малий розмір. Із заявленим розміром 150 МБ VMware рекламує свій ESXi як найменший у світі гіпервізор. Крім простоти обслуговування, невелика площа часто призводить до зменшення поверхні атаки зовнішніх загроз. У результаті частота виправлень ESXi зменшилася;

– Зручний монтаж. Невеликі розміри виробу сприяють швидшій установці та створенню інфраструктури. Можна завантажити ESXi за допомогою флешки.;

– Функція адміністрування ESXi включає вбудований браузер HTML5. Компанії, яким потрібні автоматизовані операції, також можуть використовувати vSphere CLI для віддаленого керування та інтерфейси прикладного програмування (API) на основі REST;

– Безпечний дизайн. Вбудоване шифрування захищає дані, що зберігаються у ваших віртуальних машинах. Коли ви додаєте доступ на основі ролей і можете реєструвати/прогнозувати все, що забажаєте, у вас є безпечна віртуальна платформа, починаючи з самого початку;

– Масштабована надійність. Незалежно від того, які програми ви запускаєте, один гіпервізор ESXi здатний обслуговувати до 128 віртуальних процесорів і 120 пристроїв на 6 ТБ пам'яті. На додаток до високопродуктивної кластерної файлової системи, ESXi використовує файлову систему віртуальної машини (VMFS), яка може виділяти більше ресурсів зберігання навіть за обмеженого фізичного простору. Віртуальні машини можуть споживати кілька ЦП одночасно;

– Широка підтримка та сумісність. Завдяки широкому застосуванню як корпоративної платформи ESXi широко підтримується постачальниками апаратного забезпечення та партнерами додатків, а також має сумісність з різними програмами та гостьовими операційними системами.

– Невеликий розмір ESXi дозволяє йому працювати на головній машині з мінімальними вимогами, як:

– Два ядра ЦП;

– Один 64-розрядний процесор x86, випущений після вересня 2006 року;

– ГБ оперативної пам’яті, гігабітний мережевий адаптер і накопичувач на 4 ГБ.

Найкращі конфігурації хостів, включаючи подвійні гігабітні або швидші адаптери Ethernet і запасні диски, повинні мати принаймні 8 ГБ оперативної пам’яті та чотири або більше ядер ЦП. Сервер ESXi повинен мати апаратну віртуалізацію для процесорів x64 у вашій організації, особливо якщо вони використовувати 64-розрядні віртуальні машини. Крім того, біти NX/XD на цих процесорах вимагають завантаження BIOS. Термін NX використовується для позначення Never eXecute, який сумісний лише з процесорами AMD, і перевизначення версії Intel як опції Disable. Зони зберігання обладнання захищені від крадіжки за допомогою біта NX/XD, який зберігає пам’ять для зловмисних атак. Це додає ще один рівень безпеки серверу ESXi.

Інші вимоги полягають у диску SCSI або RAID із налаштованим номером логічного блоку та нерозподіленим простором для розміщення віртуальних машин. Також підтримуються накопичувачі Serial ATA (SATA). Жорсткі диски, компакт-диски та носії USB можна завантажувати з уніфікованого розширюваного інтерфейсу прошивки (UEFI) на хостах ESXi. Також підтримується надання хостів ESXi через UEFI та мережеве завантаження. ESXi також може працювати на дисках ємністю 2 ТБ або більше для сумісного обладнання.

Встановлення VMware ESXI включає наступні кроки

Крок 1. Початок створення нової віртуальної машини в VMware ESXI, зображено на рис. 2.12.

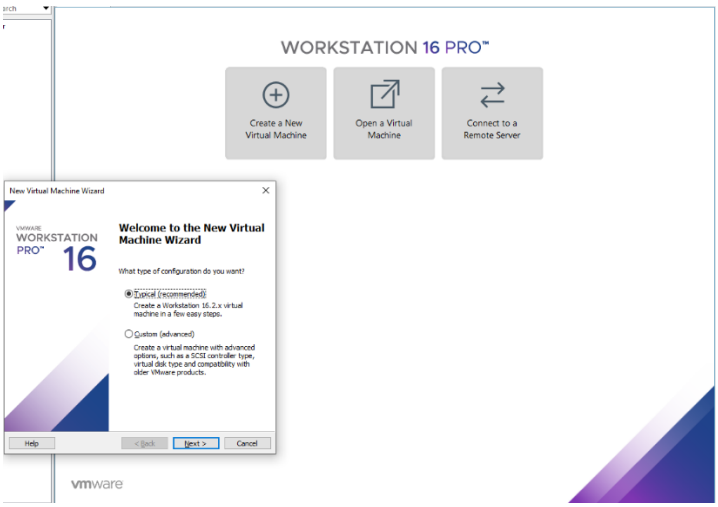


Рисунок 2.12 – Початок створення нової віртуальної машини

Крок 2. Знайдіть образ VMware ESXI і натисніть «Далі» у наступному вікні, зображено на рис. 2.13.

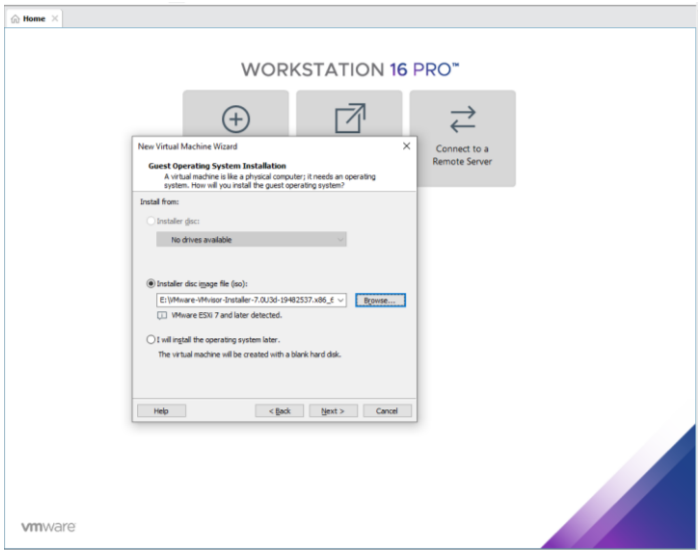


Рисунок 2.13 – Шлях до образу ОС

Крок 3. Надаємо ім’я та місце, де його можна буде розташувати.

Крок 4. Надайте інформацію про тип диска та його розмір. Починаємо встановлення VMware ESXI, зображено на рис. 2.14, 2.15.

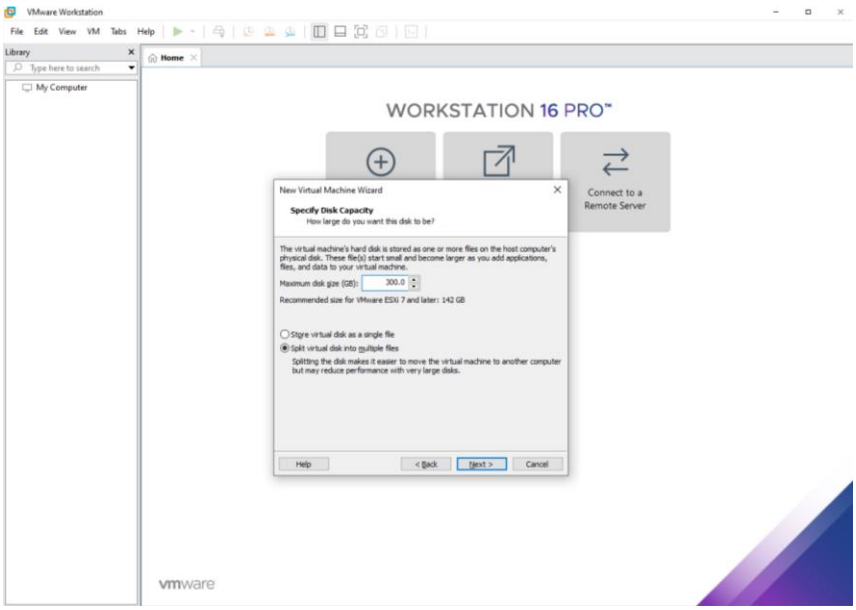


Рисунок 2.14 – Розмір жорсткого диску для віртуальної машини

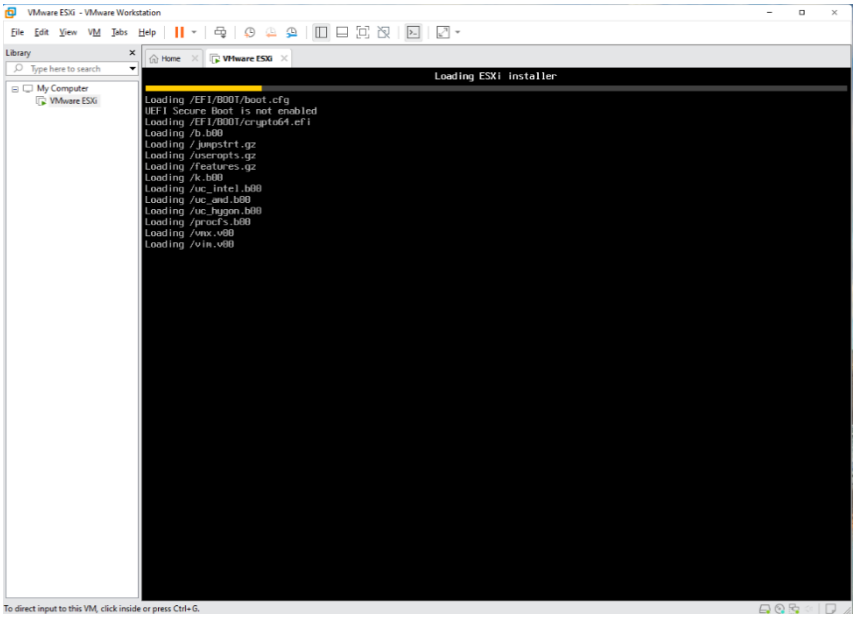


Рисунок 2.15 – Встановлення VMware ESXI

Крок 5. Сторюємо тв вводимо пароль, бажано щоб пароль був від 15 символів та містив у собі: букви, цифри, та символи.

Крок 6. При закінчені встановлення VMware ESXI на робочому вікні буде показано ip-адрес(адрес нашого сервера) який нам віділив DHCP-сервер до якого був підключен VMware ESXI . Цей ip-адрес у будь який момент можно змінити, зображено на рис. 2.16.

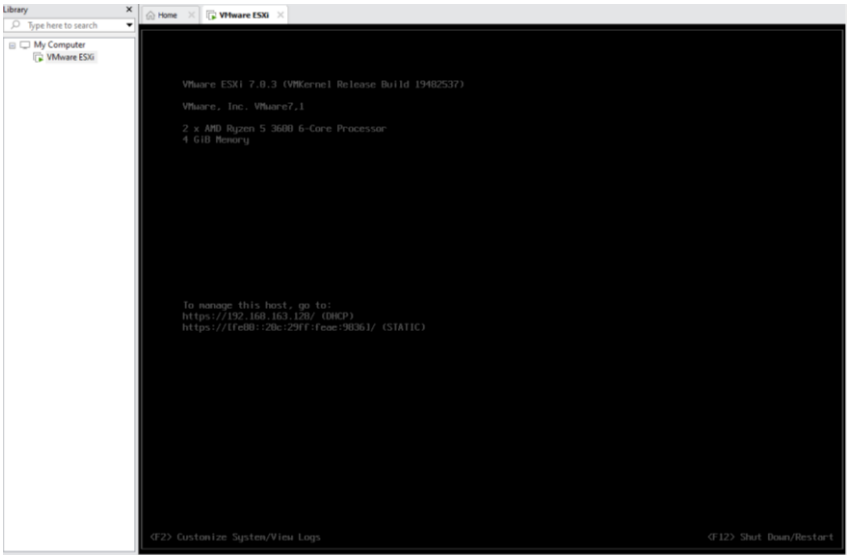


Рисунок 2.16 – Встановлена VMware ESXI

Крок 7. Після вводу ip-адреси у браузері ми зможемо отримати доступ до графічного налаштуваня серверу-віртуалізації, зображено на рис. 2.17.

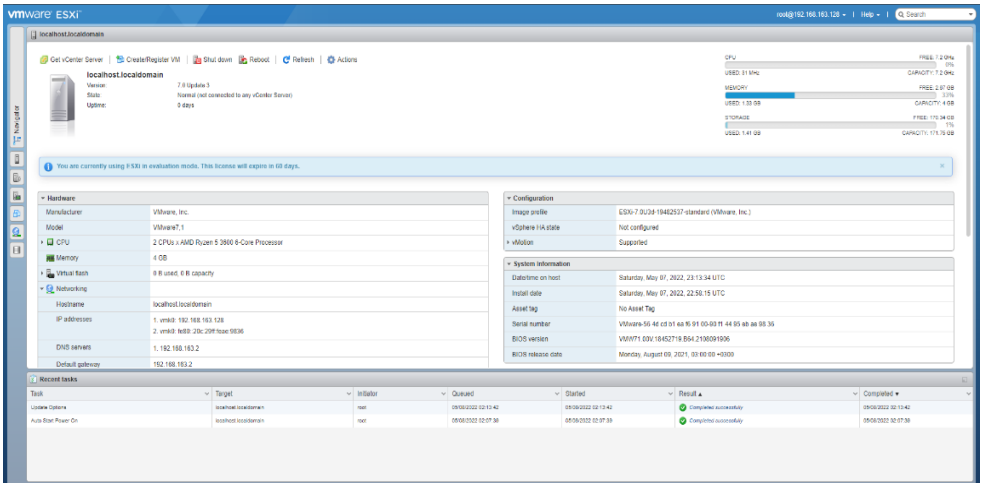


Рисунок 2.17 – Головний екран VMware ESXI

Ми можемо створити кілька віртуальних машин і налаштувати їх відповідно до потреб користувачів, а також забезпечувати віддалений доступ до віртуальної машини після встановлення VMware ESXi. Завдяки цьому є багато переваг створення нових робочих місць;

– Закупівля нових комлектуючих

– Надання простору для нового робочого місця

– Фізична присутність для її обслуговування

– Фізична присутність працівника робочого місця

2.2.1 Створення віртуальної машини за допомогою VMware ESXI

Щоб створити віртуальну машину у VMware ESXi, необхідно встановити програму. VMware ESXi — це гіпервізор типу 1, тобто його потрібно інсталювати на «голій» машині, хоча його можна розмістити в операційній системі (робочій машині) з деякими ОС. Існують два способи встановити VMware ESXI:

– Використання гіпервізорів типу 2, таких як VMware Workstation і Oracle VM VirtualBox, є варіантом встановлення VMware ESXI на робочій машині, яка вже має операційну систему. Цей спосіб вказан у розділі 2.2;

– Щоб інсталювати VMware ESXi на самому апаратному забезпеченні, необхідно створити мультимедійний образ. Потім буде створено образ, і ви зможете інсталювати VMware ESXi як звичайну ОС.

Після встановлення VMware ESXi і доступу до головного екрана ми можемо продовжити налаштування нової віртуальної машини та встановлення на ній операційної системи. Цей процес займає лише декілька кроків.

Крок 1. Натискаємо "Create / Register VM".Зображено на рис. 2.18.

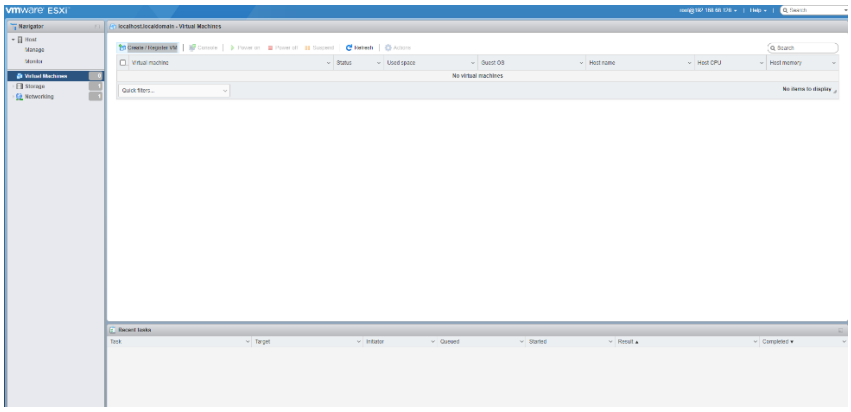


Рисунок 2.18 – Сворення віртуальної машини у VMware ESXI

Крок 2. Обираємо тип віртуальної машини.

Крок 3. Вказуємо ім'я та тип ОС яка буде встановлена.

Крок 4. Обираємо сховище.

Крок 5. Налаштовуємо віртуальну машину.Натискаємо "Datastore ISO file" та завантажуємо ОС яка буде встановлена. Зображено на рис. 2.19

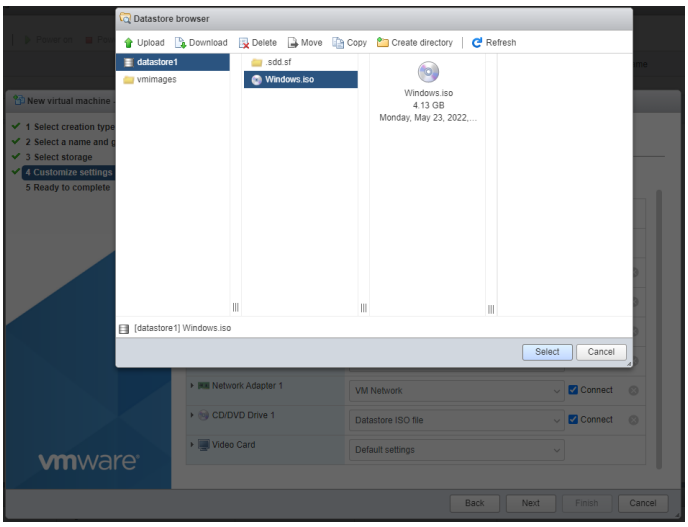


Рисунок 2.19 – Завантаження ОС

Крок 6. Натискаємо "Finish" та створюємо нову віртуальну машину. На рис 2.20 показано налаштування віртуальної машини.

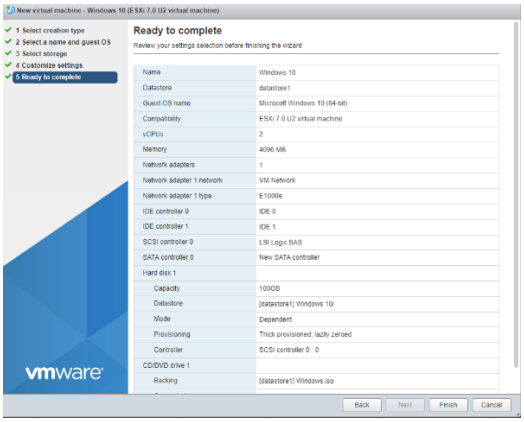


Рисунок 2.20 – Налаштування віртуальної машини

Крок 7. Запускаємо віртуальну машину та встановлюємо ОС. Результат зображено на рис. 2.21

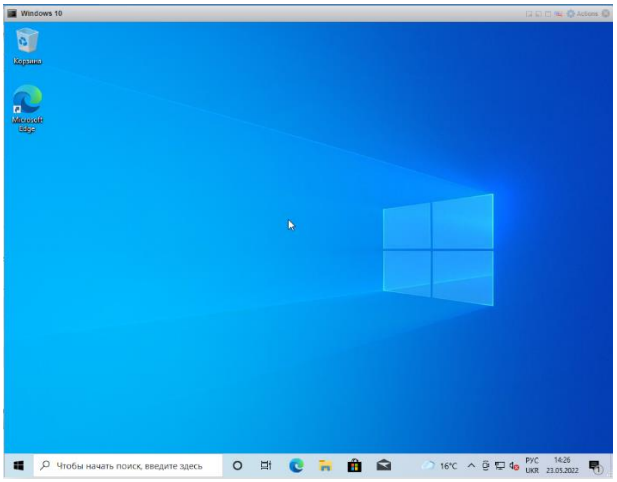


Рисунок 2.21– Віртуальна машина

РОЗДІЛ 3 РЕАЛІЗАЦІЯ СУЧАСНИХ ТЕХНОЛОГІЙ ВІРТУАЛІЗІЦІЇ НА БАЗІ ГІПЕРВІЗОРА «PROXMOX»

3.1 Гіпервізор «Proxmox»

Віртуальне середовище Proxmox — це потужна платформа віртуалізації сервера з відкритим кодом для керування двома технологіями віртуалізації — KVM (віртуальна машина на основі ядра) для віртуальних машин і LXC для контейнерів — за допомогою єдиного веб-інтерфейсу. Він також інтегрує готові інструменти для налаштування високої доступності між серверами, програмно визначеним сховищем, мережею та аварійним відновленням.

Віртуальне середовище Proxmox базується на Debian GNU/Linux і використовує спеціальне ядро Linux. Вихідний код Proxmox VE безкоштовний, випущений згідно з GNU Affero General Public License, v3 (GNU AGPLv3). Це означає, що ви можете вільно використовувати програмне забезпечення, перевіряти вихідний код у будь-який час і робити свій внесок у проект. Використання програмного забезпечення з відкритим кодом гарантує повний доступ до всіх функцій, а також високий рівень надійності та безпеки. Особливими рисами Proxmox є :

– Віртуальна машина на основі ядра (KVM). KVM — це провідна в галузі технологія віртуалізації Linux для повної віртуалізації. Це модуль ядра, який об’єднано в основне ядро Linux, і він працює з майже рідною продуктивністю на всьому обладнанні x86 із підтримкою віртуалізації — Intel VT-x або AMD-V. За допомогою KVM ви можете запускати як Windows, так і Linux у віртуальних машинах (ВМ), де кожна ВМ має приватне віртуалізоване обладнання: мережеву карту, диск, графічний адаптер тощо. Запуск кількох програм у ВМ в одній системі дає змогу заощадити потужності та зниження витрат, водночас надаючи вам гнучкість для створення гнучкого та масштабованого програмно-визначаного

центру обробки даних, який відповідає вимогам вашого бізнесу. Proxmox VE включає підтримку KVM з самого початку проекту, ще в 2008 році;

– Контейнери Linux (LXC). LXC — це середовище віртуалізації на рівні операційної системи для запуску кількох ізольованих систем Linux на одному керуючому хості Linux. LXC працює як інтерфейс простору користувача для функцій обмеження ядра Linux. Користувачі можуть легко створювати та керувати системними контейнерами або контейнерами додатків за допомогою потужного API та простих інструментів;

– Центральне управління. Щоб керувати всіма завданнями вашого віртуального центру обробки даних, ви можете використовувати центральний веб-інтерфейс керування. Уся функціональність веб-інтерфейсу також доступна через CLI або REST API, які можна використовувати для автоматизації завдань;

– Веб-інтерфейс керування. Proxmox VE простий у використанні. Ви можете виконувати всі завдання керування за допомогою інтегрованого графічного інтерфейсу користувача (GUI), немає необхідності встановлювати окремий інструмент керування. Центральний веб-інтерфейс заснований на фреймворку ExtJS JavaScript і доступний з будь-якого сучасного браузера. Окрім завдань керування, він також надає огляд історії завдань і системних журналів кожного вузла. Це включає виконання завдань резервного копіювання, оперативну міграцію, програмно визначене сховище або дії, ініційовані високою доступністю. Інструмент із кількома майстрами дозволяє керувати всім кластером з будь-якого вузла кластера; вам не потрібен виділений вузол менеджера;

– Proxmox VE мобільний. Ви можете отримати доступ до Proxmox VE на мобільних пристроях через додаток для Android або через мобільну версію веб-інтерфейсу на основі HTML5. Додаток Proxmox VE для Android базується на фреймворку Flutter і дозволяє отримати доступ до вашого сервера Proxmox VE та керувати кластером, вузлами, віртуальними машинами та контейнерами. Мобільний клієнт Proxmox VE HTML5 дозволяє вам керувати Proxmox VE на ходу,

включаючи доступ до консолі SPICE і HTML5. Це дозволяє керувати віртуальними машинами та контейнерами, а також переглядати їх конфігурацію;

– Інтерфейс командного рядка (CLI). Для досвідчених користувачів, які звикли до комфорту оболонки Unix або Windows Powershell, Proxmox VE надає інтерфейс командного рядка для керування всіма компонентами вашого віртуального середовища. Цей інтерфейс командного рядка має інтелектуальне завершення вкладок і повну документацію у формі довідкових сторінок UNIX;

– REST API. Proxmox VE використовує RESTful API. JSON вибран як основний формат даних, і весь API формально визначено за допомогою схеми JSON. Це забезпечує швидку та просту інтеграцію сторонніх інструментів керування, таких як спеціальні середовища хостингу;

– Кластеризація. Хоча багато людей починають з одного вузла, віртуальне середовище Proxmox може масштабуватися до великого набору кластерних вузлів. Стек кластера повністю інтегрований і постачається разом із інсталяцією за замовчуванням. Proxmox VE використовує унікальну кластерну файлову систему Proxmox (pmxcfs) , файлову систему, керовану базою даних, розроблену Proxmox. Pmxcfs дає змогу синхронізувати файли конфігурації у вашому кластері. За допомогою Corosync ці файли копіюються в реальному часі на всі вузли кластера. Файлова система зберігає всі дані в постійній базі даних на диску, але копія даних зберігається в оперативній пам’яті. Максимальний розмір пам’яті на даний момент становить 30 МБ — цього більш ніж достатньо для зберігання конфігурації кількох тисяч віртуальних машин. Proxmox VE — єдина платформа віртуалізації, яка використовує цю унікальну файлову систему кластера pmxcfs;

– Жива/онлайн міграція. За допомогою інтегрованої функції міграції в реальному часі/онлайн ви можете переміщувати запущені віртуальні машини з одного вузла кластера Proxmox VE на інший без будь-яких простоїв або помітного ефекту з боку кінцевого користувача. Адміністратори можуть ініціювати цей процес або з веб-інтерфейсу, або з командного рядка. Це дає змогу мінімізувати час простою, якщо вам потрібно відключити хост-систему для обслуговування;

– Унікальний мульти-майстер дизайн. Щоб спростити керування кластером, ви можете виконувати завдання обслуговування в усьому кластері з будь-якого вузла. Інтегрований веб-інтерфейс керування надає чіткий огляд усіх гостьових систем KVM і контейнерів Linux у вашому кластері. Ви можете легко керувати своїми віртуальними машинами та контейнерами, сховищами чи кластерами з графічного інтерфейсу. Немає необхідності встановлювати окремий, складний і дорогий сервер керування;

– Адміністрування на основі ролей. Ви можете визначити детальний доступ до всіх об’єктів (наприклад, віртуальних машин, сховищ, вузлів тощо) за допомогою системи керування дозволами на основі ролей. Це дозволяє визначати привілеї та допомагає контролювати доступ до об’єктів. Ця концепція також відома як списки контролю доступу: кожен дозвіл визначає тему (групу користувачів або маркер API) і роль (набір привілеїв) на певному шляху. Proxmox VE підтримує кілька джерел автентифікації, наприклад Linux PAM, інтегрований сервер автентифікації Proxmox VE, LDAP, Microsoft Active Directory і OpenID Connect;

– Кластер високої доступності (HA) Proxmox VE. Багатовузловий кластер Proxmox VE дозволяє створювати високодоступні віртуальні сервери. Кластер Proxmox VE HA базується на перевірених технологіях високої доступності Linux, що забезпечує стабільну та надійну службу високої доступності. Весь кластер Proxmox VE HA можна легко налаштувати за допомогою вбудованого веб-інтерфейсу користувача;

– Менеджер Proxmox VE HA. Менеджер ресурсів, Proxmox VE HA Manager, відстежує всі віртуальні машини та контейнери в кластері та автоматично починає діяти, якщо один із них виходить з ладу. Proxmox VE HA Manager працює з коробки. Потрібна нульова конфігурація. Крім того, огородження на основі сторожового таймера значно спрощує розгортання;

– Симулятор Proxmox VE HA. Proxmox VE містить симулятор високої доступності. Це дозволяє перевірити поведінку реального кластера з 3 вузлами та

6 віртуальними машинами. Симулятор Proxmox HA працює з коробки та допомагає вам дізнатися та зрозуміти, як працює Proxmox VE HA;

– Програмно-визначена мережа (SDN). Мережеві можливості Proxmox VE дозволяють створювати чутливі та адаптовані мережі, які можна масштабувати відповідно до потреб вашого бізнесу. Функція програмно-визначеної мережі (SDN) дозволяє Proxmox VE керувати розширеними мережевими конфігураціями та налаштуваннями мультиарендації в кластерах Proxmox VE. Можливі варіанти використання варіюються від ізольованої приватної мережі на кожному окремому вузлі до складних накладених мереж у кількох кластерах Proxmox VE у різних місцях. Він застосовний до мереж будь-якого розміру, від простої маршрутизації NAT, традиційного поділу на 802.1q VLAN до таких функцій, як QinQ, тунелювання VXLAN та інфраструктури EVPN на основі BGP;

– Мережевий стек Linux. Для простих налаштувань Proxmox VE надає гнучкі параметри конфігурації мережі для локальних вузлів. Модель мостової мережі використовується для забезпечення безперебійного з’єднання між віртуалізованими середовищами та зовнішньою мережею. Мости можна порівняти з фізичними мережевими комутаторами, але реалізовані як програмне забезпечення на хості Proxmox VE. Для додаткової гнучкості ви можете налаштувати VLAN, створити інтерфейс зв’язку та керувати базовою мережевою маршрутизацією.

– Гнучкі можливості зберігання. Модель зберігання Proxmox VE дуже гнучка. Образи віртуальних машин можна зберігати на одному або кількох локальних пристроях зберігання або на спільному сховищі, наприклад NFS і SAN. Обмежень немає. Ви можете налаштувати скільки завгодно сховищ і використовувати всі технології зберігання, доступні для Debian GNU/Linux. Перевагою зберігання віртуальних машин у спільному сховищі є можливість миттєвої міграції запущених машин без будь-яких простоїв.У веб-інтерфейсі Proxmox VE ви можете додати такі типи зберігання: LVM Group, iSCSI target, NFS Share, SMB/CIFS, Ceph RBD, Direct to iSCSI LUN, GlusterFS, CephFS, LVM Group, Directory ZFS;

– Програмно-визначене сховище з Ceph. Ceph — це сховище розподілених об’єктів і файлова система з відкритим вихідним кодом, розроблена для забезпечення чудової продуктивності, надійності та масштабованості. Віртуальне середовище Proxmox повністю інтегрує Ceph, надаючи вам можливість запускати та керувати сховищем Ceph безпосередньо з будь-якого вузла вашого кластера. Ceph надає два типи сховища: RADOS Block Device (RBD) і CephFS. RBD забезпечує зберігання на рівні блоків для такого вмісту, як образи дисків і знімки. CephFS реалізує POSIX-сумісну файлову систему, використовуючи для зберігання своїх даних кластер зберігання Ceph. Переваги Ceph з Proxmox VE:

– Просте налаштування та керування через GUI та CLI;

– Самовідновлення;

– Масштабується до ексабайтного рівня;

– Налаштуйте пули з різними характеристиками продуктивності та резервування;

– Працює на економічному товарному обладнанні.

– Брандмауер Proxmox VE. Вбудований брандмауер Proxmox VE забезпечує простий спосіб захисту вашої IT-інфраструктури. Брандмауер повністю настроюється, дозволяючи складні конфігурації через GUI або CLI. Ви можете встановити правила брандмауера для всіх хостів усередині кластера або визначити правила лише для віртуальних машин і контейнерів. Такі функції, як макроси брандмауера, групи безпеки, набори IP-адрес і псевдоніми, допомагають полегшити це завдання;

– Розподілений міжмережевий екран. Хоча вся конфігурація зберігається у файловій системі кластера, брандмауер на основі iptables працює на кожному вузлі кластера і, таким чином, забезпечує повну ізоляцію між віртуальними машинами. Розподілений характер цієї системи також забезпечує набагато вищу пропускну здатність, ніж централізоване рішення брандмауера. Брандмауер має повну підтримку IPv4 та IPv6. Підтримка IPv6 є повністю прозорою, і ми фільтруємо

трафік для обох протоколів за замовчуванням. Таким чином, немає необхідності підтримувати інший набір правил для IPv6;

– Резервне копіювання/відновлення. Резервне копіювання є основною вимогою для будь-якого розумного IT-середовища. Платформа Proxmox VE забезпечує повністю інтегроване рішення, використовуючи можливості кожного сховища та кожного типу гостьової системи. Резервне копіювання можна легко запустити за допомогою графічного інтерфейсу користувача або інструменту резервного копіювання vzdump (через командний рядок). Ці резервні копії завжди є повними резервними копіями, які містять конфігурацію віртуальних машин і контейнера, а також усі дані. Інтегрований інструмент резервного копіювання (vzdump) створює узгоджені знімки запущених контейнерів і гостьових систем KVM. По суті, він створює архів даних віртуальної машини або контейнера, а також містить файли конфігурації. Завдання резервного копіювання можна запланувати таким чином, щоб вони автоматично виконувалися в певні дні та час для вибраних вузлів і гостьових систем. Живе резервне копіювання KVM працює для всіх типів сховищ, включаючи образи віртуальних машин на NFS, iSCSI LUN і Ceph RBD. Формат резервного копіювання Proxmox VE оптимізований для швидкого та ефективного зберігання резервних копій віртуальної машини (враховуючи розріджені файли, невпорядковані дані, мінімізоване введення/виведення).

– Інтеграція сервера резервного копіювання Proxmox. Proxmox Backup Server — це рішення для резервного копіювання корпоративного класу, яке здатне створювати резервні копії віртуальних машин, контейнерів і фізичних хостів. Підтримка цього повністю інтегрована в Proxmox VE, тобто ви можете легко створювати резервні копії та відновлювати гостьові дані за допомогою того самого загального інтерфейсу, який використовують інші типи сховищ. Ці резервні копії є додатковими, лише передають нещодавно змінені дані через мережу. Це дуже вигідно з точки зору пропускної здатності мережі та часу виконання завдання резервного копіювання. Дані також можна легко зашифрувати на стороні клієнта, щоб ваші резервні копії даних були недоступні

для зловмисників. Відновлення великих резервних копій може зайняти тривалий час і стати основною причиною простою в разі аварії. Однак для резервних копій віртуальної машини, які зберігаються на сервері резервного копіювання Proxmox, функція живого відновлення мінімізує час простою, дозволяючи віртуальній машині запускатися, як тільки почнеться відновлення. У цьому випадку дані постійно копіюються у фоновому режимі, встановлюючи пріоритет фрагментам, до яких віртуальна машина активно звертається. Часто з резервної копії потрібен лише один файл або каталог. За допомогою веб-інтерфейсу Proxmox VE ви можете безпечно шукати та відновлювати окремі файли чи каталоги з резервної копії віртуальної машини або контейнера.

3.2 Встановлення Proxmox

Для виробництва серверів необхідне високоякісне серверне обладнання. Proxmox VE підтримує кластеризацію, це означає, що кількома інсталяціями Proxmox VE можна централізовано керувати завдяки інтегрованій функціональності кластера. Proxmox VE може використовувати локальне сховище (DAS), SAN, NAS, а також спільне та розподілене сховище (Ceph). Для того щоб сервер з віртуалізацією міг нормально працювати для машини на якому буде встановлений гіпервізор повинен відповідати мінімальним вимогам:

– Intel EMT64 або AMD64 із прапорцем процесора Intel VT/AMD-V;

– Пам'ять, мінімум 2 ГБ для ОС і служб Proxmox VE. Плюс виділена пам'ять для гостей. Для Ceph або ZFS потрібна додаткова пам’ять, приблизно 1 ГБ пам’яті на кожен використаний ТБ пам’яті;

– Для швидкого резервного зберігання, найкраще мати SSD-дисками;

– Зберігання ОС: апаратний RAID із кеш-пам’яттю запису, захищеним від батарей («BBU») або без RAID із кеш-пам’яттю ZFS і SSD;

– Сховище віртуальної машини: для локального зберігання використовуйте апаратний RAID з кеш-пам’яттю запису, що живиться від батареї (BBU) або без RAID для ZFS. Ані ZFS, ані Ceph несумісні з апаратним контролером RAID. Також можливе спільне та розподілене зберігання;

– Також підтримуються резервні мережеві карти Гбіт, додаткові мережеві карти залежно від бажаної технології зберігання та налаштування кластера 10 Гбіт і вище;

– Для проходження PCI(e) потрібен ЦП із прапором ЦП VT-d/AMD-d.

Для встановлення гипервізора Proxmox на робочій машині вам потрібно мати встановлючий носій який ви можете зробити завантавжиши iso файл з офіційного сайту розробника та створити його за допомогою будь якого додатка який може створювяти такі носії. Перед початков встановлення Proxmox рекомендуємо перевірити налаштування BIOS саме такі параметри :

– Intel (VMX) Virtualization Technology для материнських плат від asus та процесорів Intel. Зазвичай вони знаходятся за таким шляхом: «Advanced Mode», «Advanced», «CPU Configuration»;

– SMV для материнських плат від gigabyte яке знаходится у вікні налаштування M.I.T.

Після зміни налаштувань та запуску робочої машини з підключеним до неї носія нам потрібно зробити декілька простих етапів.

Етап 1. Вибрати Install Proxmox VE.

Етап 2. Далі ви побачите ліцензійну угоду. Щоб продовжити прійміть її.

Етап 3. Виберіть диск, на який буде інстальовано систему. Якщо потрібно змінити параметри диска, натисніть «Options» і внесіть налаштування.

Етап 4. Виберіть країну, часовий пояс та розкладку клавіатури.

Етап 5. Вкажіть пароль суперкористувача та email для повідомлень.

Етап 6. Виберіть інтерфейс мережі. Введіть ім'я хоста, IP-адресу, маску підмережі, шлюз та DNS Server. Якщо ваша машина підключена до мережі з DHCP-сервером це все визначится автоматично.

Етап 7. Дочекайтеся закінчення установки.

Після проходження всіх етапів встановлення та налаштування гіпервізора ви зможете скористатися web-інтерфейсом, відкрийте браузер і введіть посилання виду https://123.123.123.123:8006/, де 123.123.123.123 — IP-адреса вашого сервера. Це посилання можна побачити при запуску терміналу. Вас попросить вести логін та пароль суперкористувача де логін. Також тут ви можете обрати мову веб-інтерфейсу. Результат встановленого гіпервізора зображено на рис. 3.1.

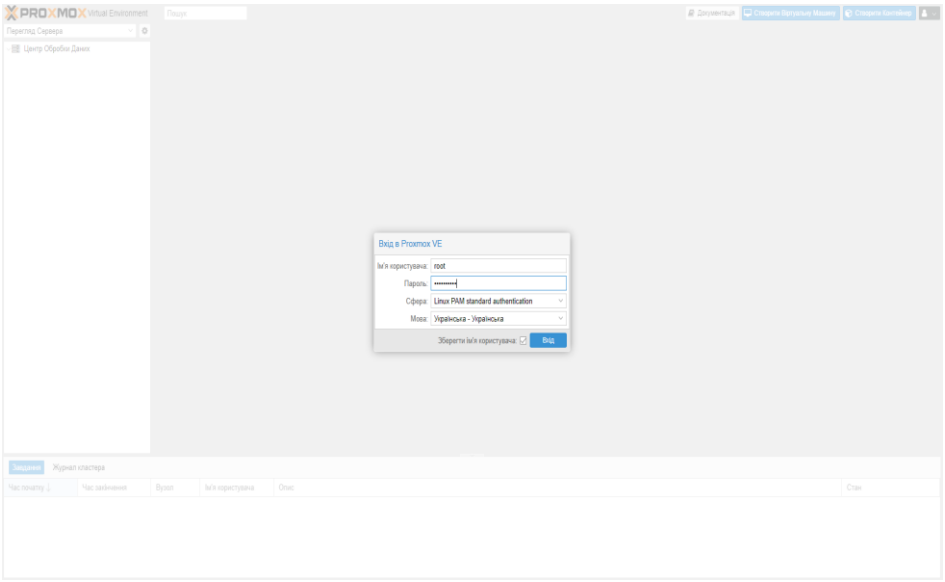


Рисунок 3.1– Веб-інтерфес Proxmox

3.3 Створення кластеру

Комп'ютерний кластер або просто клаастер — це декілька незалежних обчислювальних машин, що використовуються спільно і працюють як одна система для вирішення тих чи інших задач, наприклад, для підвищення продуктивності, забезпечення надійності, спрощення адміністрування тощо.

Обчислювальний кластер потрібен для збільшення швидкості обрахунків за допомогою паралельних обчислень.

Для створення кластеру потрібно обрати «Центр Обробки Даних», «Кластер» та натиснути «Створити Кластер».Зображено на рис. 3.2.

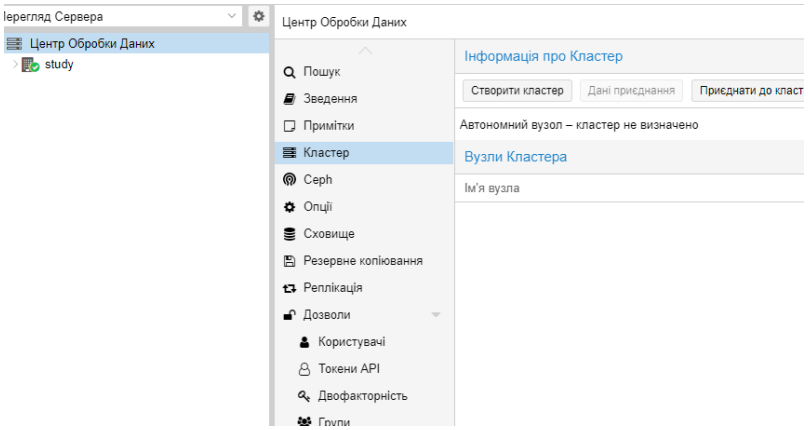


Рисунок 3.2– Створення кластеру

Далі на потрібно визначти ім’я кластеру та обрати мережу в якій вона буде створена.Якщо ваша машина знаходится у декількох мережах, для підвищення відмовостійкості кластеру, ви можете додати додаткові канали мережі. Пріорететом буде канал з найменшим числом. Зображено на рис. 3.3.

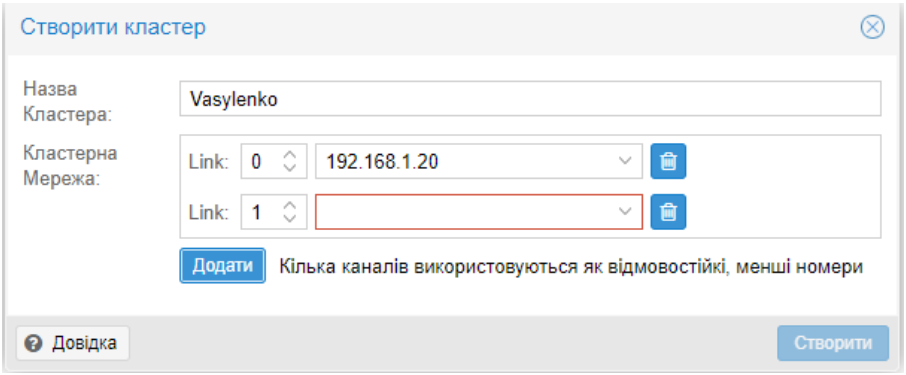


Рисунок 3.3– Налаштуання кластеру

Після створення початкового кластеру ми можемо приєднати до нього інші машини. Це робится у декілька кроків.

Крок 1. Скопійовуємо «Дані Приєднання».

Крок 2. Заходимо на іншу машини та натискаємо «Приєднати до Кластера».

Крок 3. Вставляємо скійовані дані та водимо пароль суперкористувача від машини де почаково був створений кластер. Результат зображений на рис. 3.4.

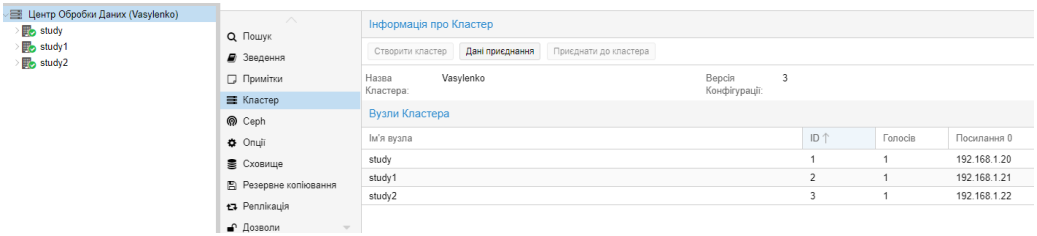


Рисунок 3.3– Створений кластер

3.4 Створення віртуальної машини

Для створення нової віртуальної машини всередині гіпервізора необхідно у верхньому правому куті web-інтерфейсу натиснути «Створити віртуальну машину», після чого відкриється вікно налаштування. Перед вами відкриється вікно налаштувань із вісьмома вкладками та безліччю параметрів. Розглянемо основні базові, ви можете змінити їх під свої потреби. Єдиний момент, який необхідно врахувати під час створення віртуальних машин, їх ресурси повинні перевищувати фізичні характеристики гипервизора. Логічно, що маючи фізично на роботі 16 Gb оперативної пам'яті, створити кілька віртуальних машин споживаючих сумарно 32 Gb у вас вийде, але при перевищенні ресурсів гіпервізор просто «піде в себе». Це ж стосується і ємності жорстких дисків. За цим важливо стежити. Зараз ми з вами розсмотремо біль детальні кроки для створення віртуальної машини.

Крок 1. Обираєм вузол нашого кластеру на якому буде створено віртуальну машину. Називаємо нашу машину та привласнюємо їй id.

Крок 2. Обираємо ISO-образ, тип та версыю нашого гостьового ОС.

Крок 3. Налаштовуємо у вікні «Система» графічну карту, контролер SCSI та BIOS. Можна залишити налаштування за замовчуванням

Крок 4. Налаштовуємо тип пристрою (диску) , сховище з якого буде взятий об’єм диску та розмір диску яке буде використовувати віртуальна машина. Також при необхідності можемо обмежити пропускну здатній віртуального диску.

Крок 5. Обираємо кількість сокетів та ядер для віртуальної машини.

Крок 6. Встановлюємо об’єм оперативної пам’яті.

Крок 7. Налаштовуємо тип та модель мережі віртуальної машини.

Крок 8. Перевіряємо налаштування нової віртуальної машини та закінчуємо її створеня. Результат створеної віртуальної машини зображено на рис. 3.5.

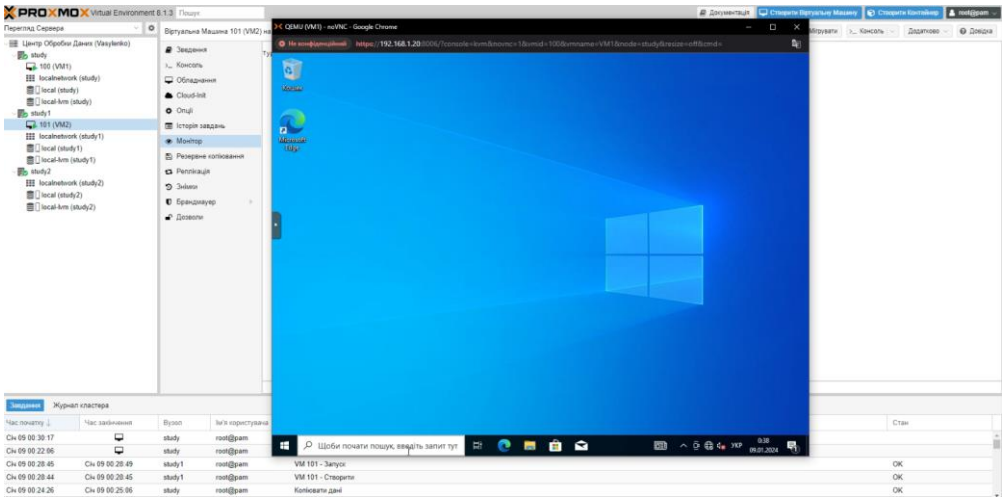


Рисунок 3.5 – Створена віртуальна машина

3.5 Висока доступність Proxmox

Висока доступність є комбінацією компонентів і налаштувань, які уможливлюють безперервну роботу обчислювального середовища протягом тривалого часу на регулярній основі. В основному це означає, що навіть коли обладнання сервера, що знаходиться в автоматичному режимі, має проблеми в середовищі реального часу, висока доступність (HA) може керувати серверами, що залишилися самостійно і підтримуючи віртуальне середовище в робочому стані автоматично переміщуючи або виконуючи міграцію віртуальних машин з одного

вузла на інший . Налаштована належним чином HA вимагає дуже незначного реального втручання користувачів у разі відмови апаратних засобів. Без HA на своєму місці, всі вузли вимагають постійного моніторингу з боку мережевих менеджерів, щоб вручну переміщати віртуальні машини на життєздатні вузли коли деякі вузли мають проблеми.

У невеликих середовищах переміщення вручну ВМ не є головною проблемою, проте у великих середовищах із сотень віртуальних машин чи вузлів постійний моніторинг може бути дуже затратним у сенсі часу. Незважаючи на те, що в системі може існувати програмне забезпечення моніторингу, без HA адміністратор буде повинен вручну переміщати або виконувати міграцію будь-яких віртуальних машин з вузла, що відмовив. Це може спричинити значний час простою через час відгуку мережевого персоналу. Це саме те місце, де набуває чинності функціональність HA Proxmox. HA виводить втручання оператора за дужки рішення, просто переміщуючи або виконуючи міграцію віртуальних машин, як тільки виникає відмова обладнання сервера.

Для встановлення функціональності HA в Proxmox важливо мати всі ваші віртуальні машини на сховищі, що спільно використовується. Дуже важливо розуміти, що HA Proxmox обробляє лише вузли Proxmox та віртуальні машини в межах кластера Proxmox. Таку функціональність HA не слід плутати з надмірністю спільно використовуваних сховищ, яку Proxmox може застосовувати у своєму розгортанні HA. Висока доступність у спільному сховищі просто так само важлива, як і висока доступність ВМ Proxmox. Спільно використовувані сховища сторонніх виробників можуть надавати власну функціональність HA. Таким чином, і сам кластер Proxmox, і сховище, що спільно використовується, повинні бути налаштовані для надання реального середовища з високою доступністю.

Можуть існувати рівні надмірності в обчислювальному вузді Proxmox, такі як застосування RAID, надлишкові джерела живлення, агреговані мережеві зв'язки або зчеплення (bond). HA в Proxmox не підміняє собою жодного з цих рівнів. Він

просто сприяє використанню функцій надмірності віртуальними машинами для збереження їх у робочому стані в процесі відмови будь-якого вузла.

Слід зазначити, що у вузлі Proxmox викликана необхідністю застосування оновлень перезавантаження викликає вимкнення всіх віртуальних машин із включеної HA з переміщенням їх на наступний доступний вузол Proxmox та їх подальший повторний запуск. У подібній ситуації може виявитися необхідною міграція в реальному часі вручну віртуальних машин до перезавантаження вузла, що оновлюється.

У випадку, коли з якоїсь причини вузол стає недоступним, HA Proxmox очікує 60 секунд перш ніж виконається огорожа (fencing) вузла, що відмовив. Огородження запобігає службам кластера від повернення в робочий стан у цьому місці. Потім HA переміщує ці ВМ на наступний доступний вузол у групі учасників HA. У Proxmox VE 4.1 контейнери LXC не можуть виконувати міграцію в реальному часі. Тому HA зупинить всі LXC контейнери, а потім перемістить їх на наступний доступний вузол. Навіть якщо вузол з ВМ все ще включений, але втратив зв'язок із мережевим середовищем, HA Proxmox спробує перемістити всі ВМ із цього вузла на інший вузол.

Однак, коли вузол, що відмовив, повертається в робочий стан, HA не буде автоматично переміщати ВМ на початковий вузол. Це потрібно виконувати вручну. Однак, ВМ може бути переміщена вручну тільки якщо HA заборонений для такої ВМ. Тому ми спочатку повинні вимкнути HA, а потім перемістити на початковий вузол і включити HA на цій ВМ знову. Як ми можемо бачити, HA Proxmox любить виконувати все управління на свій розсуд, хоч і залишаючи невеликі неприємності для виконання певних функцій вручну. HA орієнтований підтримку безвідмовної роботи, яка є завданням, виконуваним їм належним чином. Далі в цьому розділі ми розглянемо, як налаштовувати HA для віртуальних машин.

Починаючи з Proxmox 4.0 і для пізніших версій, функціональність високої доступності була повністю перебудована починаючи з основи, що зробило її більш

простою в налаштуванні та застосуванні. Існує кілька вимог, яким має відповідати середовище перед налаштуванням HA Proxmox. Вони такі:

– Мінімально потрібно три вузли;

– Спільно використовуване сховище;

– Огородження (fencing).

HA повинна налаштовуватися в кластері, що містить як мінімум три вузли, оскільки три вузли і більше можуть досягати Кворуму. Кворум є мінімальною кількістю голосів, необхідних для кластера Proxmox. Це мінімальне значення є загальною кількістю голосів більшості наявних вузлів. Наприклад, у кластері з трьох вузлів Proxmox, для формування Кворуму потрібно два вузли Proxmox. Або в кластері з восьми вузлів для отримання Кворуму потрібно п'ять вузлів Proxmox. За наявності двох вузлів співвідношення голосів залишається 1:1, тому ніякий кворум неможливий.

При відмові вузла ВМ переміщаються на наступний вузол учасник у HA. Подібна міграція зазвичай відбувається у реальному часі. ВМ не може виконувати міграцію в реальному часі, коли образ диска не зберігається в сховищі, що спільно використовується.

Огородження (fencing) є концепцією ізоляції вузла чи його ресурсів у разі відмови якогось вузла про те, щоб вузли, що залишилися, не могли мати доступ до тих самих ресурсів, поміщаючи їх у зону ризику для руйнування даних. У Proxmox огорожа запобігає можливості виконання на безлічі вузлів однієї і тієї ж віртуальної машини або специфічної для кластера служби. Огородження забезпечує цілісність даних у процесі відмови вузла запобігаючи всім вузлам виконання однієї й тієї ж віртуальної машини чи служб кластера одночасно.

Proxmox VE 4.1 більше не потребує використання окремого пристрою огорожі (Fencing) для налаштування HA Proxmox. Огорожа тепер застосовує апаратні засоби забезпечення безпеки (watchdog) або програмні засоби Linux для її відстеження (softdog). Програмні засоби безпеки (softdog) Linux є програмною

версією традиційних систем відстеження безпеки. Більшість BIOS сучасних серверів мають функціональність відстеження, проте вона зазвичай відключена за умовчанням. Якщо вона включена, вона перезавантажує вузли сервера після певного проміжку відсутності активності. HA Proxmox завжди перевірятиме наявність апаратного відстеження, а якщо його немає, вона буде використовувати програмне відстеження. Використання програмного відстеження (softdog) тепер уможливлює для HA реалізацію вбудовування віртуального середовища. Це корисно для встановлення віртуальних середовищ Proxmox з метою вивчення та перевірки HA Proxmox без впливу змін на ваші основні системи.

Перед тим, як встановити огорожу та HA Proxmox, ми повинні переконатися, що ці вузли можуть завантажуватися негайно після виконання циклу включення або втрати живлення. За замовчуванням ця функція заборонена. Наступний знімок екрана рис. 3.6 показує цю функцію BIOS

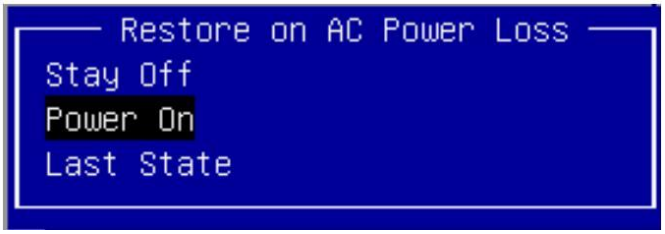


Рисунок 3.6 – Відновлення після втрати живлення

ВИСНОВКИ

Завдяки дослідженню сучасних технологій віртуалізації, які використовуються в IT-індустрії. Переваги віртуалізації включають значне зменшення кількості необхідних фізичних комп'ютерів. Таким чином, час і гроші, витрачені на пошук, придбання або заміну обладнання, скорочуються. Крім того, зменшуються площі обслуговування серверів. Для обслуговування фізичних комп'ютерів потрібно менше людей. Керівництво підприємства розцінює скорочення штату як значне зменшення витрат підприємства. Додаючи жорсткий диск або розширюючи наявний, збільшуючи обсяг оперативної пам'яті, це займає певний час у випадку з фізичним сервером. Усі кроки, необхідні для вимкнення, відключення від стійки, підключення нового обладнання та ввімкнення у віртуалізації, усуваються, що призводить до простої операції лише з 1-2 клацаннями миші або командами адміністратора. Ефективне клонування віртуальних машин є додатковою перевагою використання віртуалізації. Як приклад, підприємство урочисто відкриває однойменне приміщення. При цьому серверна інфраструктура штаб-квартири стандартизована, і існує кілька серверів з однаковими налаштуваннями. Для розгортання інфраструктури потрібно просто скопіювати зображення на сервер нового офісу, налаштувати мережеве обладнання та змінити налаштування програмного забезпечення. Хост і віртуальні машини зберігаються окремо. Розробка програмного забезпечення має переваги, які це пропонує. Це включає в себе розробку вірусних програм і програмного забезпечення, які можуть спричинити пряме втручання в операційну систему. Наша основна робоча станція не може бути заражена вірусами або зламана, тому ми використовуємо нашу ізольовану віртуальну машину для тестування програмного забезпечення.

Завдяки зручному встановленню на сервері VMware ESXi є, по суті, апаратним гіпервізором, який дозволяє працювати декільком віртуальним машинам. Використання ESXi дозволяє консолідувати кілька серверів на меншій кількості фізичних пристроїв, що призводить до більш ефективного використання

місця, зменшення енергоспоживання та підвищення продуктивності. VirtualBox для OracleVM є найпоширенішим рішенням віртуалізації для кросплатформних операційних систем у всьому світі. Oracle VM VirtualBox Enterprise дозволяє конфігурувати комп'ютери для кількох операційних систем. Розроблений для IT-фахівців, Oracle VM VirtualBox можна використовувати для розробки програмних рішень, які можуть працювати на кількох платформах на одному комп'ютері. Встановлення та робота Oracle VM VirtualBox не викликає зусиль завдяки наявності швидкого та потужного механізму віртуалізації. Proxmox буде гарним вибором для тих, хто хоче використовувати відкрите та безкоштовне програмне забезпечення з гнучкими опціями віртуалізації. Proxmox є повністю безкоштовним та відкритим програмним забезпеченням на базі Debian Linux, яке підтримує як повну віртуалізацію KVM для Windows та Linux, Proxmox надає функціонал резервного копіювання та відновлення хостів, контейнерів та віртуальних середовищ. Крім того, Proxmox підтримує автентифіковане шифрування та інкрементне резервне копіювання при рішеннях з резервного копіювання та відновлення.

Отже, у ході дослідження було з'ясовано. Що таке технологія віртуалізаціїї, її типи та переваги. Способи створення віртуальних машин та іх налаштування за допомогою гіпервізора 2 типу Oracle VM VirtualBox , VMware Workstation та 1 типу VMware ESXI, Proxmox, їх специвіки, переваг та недоліків. Також було реалізовано деякі технології віртуалізації.

Результати проведеного дослідження можуть бути рекомендовані для ознайомлення технологій віртуалізацій, створенню віртуальних машин та іх налаштування.

ПЕРЕЛІК ПОСИЛАНЬ

1. Hardware Virtualization [Електронний ресурс] – https://www.economize.cloud/glossary/hardware-virtualization

2. Hardware virtualization [Електронний ресурс] – https://en.wikipedia.org/wiki/Hardware_virtualization

3. Emulator [Електронний ресурс] – https://en.wikipedia.org/wiki/Emulator

4. Virtualization [Електронний ресурс] – https://en.wikipedia.org/wiki/Virtualization

5. Virtualization [Електронний ресурс] – https://www.ibm.com/topics/virtualization

6. Full virtualization [Електронний ресурс] – https://en.wikipedia.org/wiki/Full_virtualization

7. Full virtualization [Електронний ресурс] – https://www.sciencedirect.com/topics/computer-science/full-virtualization

8. Difference between Full Virtualization and Paravirtualization [Електронний ресурс] – https://www.geeksforgeeks.org/difference-between-full-virtualization-and-paravirtualization/

9. Application virtualization [Електронний ресурс] – https://en.wikipedia.org/wiki/Application_virtualization

10. What is Application Virtualization? [Електронний ресурс] – https://www.vmware.com/topics/glossary/content/application-virtualization.html

11. Desktop virtualization [Електронний ресурс] – https://en.wikipedia.org/wiki/Desktop_virtualization

12. What is desktop virtualization? [Електронний ресурс] – https://www.ibm.com/topics/desktop-virtualization

13. OS-level virtualization [Електронний ресурс] – https://en.wikipedia.org/wiki/OS-level_virtualization

14. Operating system based Virtualization [Електронний ресурс] – https://www.geeksforgeeks.org/operating-system-based-virtualization/

15. Virtualization: Basic concept [Електронний ресурс] – https://documentation.suse.com/smart/virtualization-cloud/html/concept-virtualization/index.html

16. Virtualization in Cloud Computing and Types [Електронний ресурс] – https://www.geeksforgeeks.org/virtualization-cloud-computing-types/

17. Oracle VM VirtualBox User Manual [Посібник] https://download.virtualbox.org/virtualbox/UserManual.pdf

18. An Overview of the Hyper-V Architecture [Електронний ресурс] – https://www.virtuatopia.com/index.php?title=An_Overview_of_the_Hyper-V_Architecture

19. Hyper-V architecture [Електронний ресурс] – https://learn.microsoft.com/en-us/windows-server/administration/performance-tuning/role/hyper-v-server/architecture

20. Virtualization Types [Електронний ресурс] – https://medium.com/@sehanitaniya8/virtualization-types-6c348cd6d999

21. Virtualization in cloud computing - The Complete Guide [Електронний ресурс] – https://www.simplilearn.com/virtualization-in-cloud-computing-article

22. 10 Benefits of Virtualization: Guide to Advance Your Business [Електронний ресурс] – https://itmagic.pro/blog/10-benefits-of-virtualization-guide-to-advance-your-business

23. What Is Virtualization? Meaning, Types, and Software [Електронний ресурс] – https://www.spiceworks.com/tech/devops/articles/what-is-virtualization/

24. What is Server Virtualization? [Електронний ресурс] – https://www.hpe.com/us/en/what-is/server-virtualization.html

25. How Server Virtualization Works [Електронний ресурс] – https://computer.howstuffworks.com/server-virtualization.htm

26. What is Server Virtualization? [Електронний ресурс] – https://www.vmware.com/topics/glossary/content/server-virtualization.html

27. What are the types of desktop virtualization [Електронний ресурс] – https://www.linkedin.com/pulse/what-types-desktop-virtualization-aniruddh-diwan

28. What is desktop virtualization? [Електронний ресурс] – https://www.citrix.com/solutions/vdi-and-daas/what-is-desktop-virtualization.html

29. VDI vs. RDS, What's the Difference? [Електронний ресурс] – https://mycloudit.com/blog/vdi-vs.-rds-whats-the-difference

30. Desktop virtualization [Електронний ресурс] – https://en.wikipedia.org/wiki/Desktop_virtualization

31. VMware Workstation [Електронний ресурс] – https://www.vmware.com/solutions/cloud-management.html

32. What Is Server Virtualization? Definition, Uses, and the Benefits [Електронний ресурс] – https://www.sangfor.com/glossary/cloud-and-infrastructure/what-is-server-virtualization

33. Proxmox Virtual Environment [Електронний ресурс] – https://uk.wikipedia.org/wiki/Proxmox_Virtual_Environment

34. How to Install Proxmox VE [Електронний ресурс] – https://phoenixnap.com/kb/install-proxmox

35. Proxmox VE Administration Guide [Електронний ресурс] – https://pve.proxmox.com/pve-docs/pve-admin-guide.html

36. Cluster Manager [Електронний ресурс] – https://pve.proxmox.com/wiki/Cluster_Manager

37. Create a Proxmox cluster [Електронний ресурс] – https://4sysops.com/archives/create-a-proxmox-cluster/

38. Qemu/KVM Virtual Machines [Електронний ресурс] – https://pve.proxmox.com/wiki/Qemu/KVM_Virtual_Machines

39. How to Create a VM in Proxmox VE [Електронний ресурс] – https://support.us.ovhcloud.com/hc/en-us/articles/360010916620-How-to-Create-a-VM-in-Proxmox-VE

40. Network File System [Електронний ресурс] – https://en.wikipedia.org/wiki/Network_File_System

41. Network File System [Електронний ресурс] – https://www.techtarget.com/searchenterprisedesktop/definition/Network-File-System

42. Storage: NFS [Електронний ресурс] – https://pve.proxmox.com/wiki/Storage:_NFS

43. High Availability Cluster [Електронний ресурс] – https://pve.proxmox.com/wiki/High_Availability_Cluster

44. High Availability [Електронний ресурс] – https://pve.proxmox.com/wiki/High_Availability

45. Storage [Електронний ресурс] – https://pve.proxmox.com/wiki/Storage

46. PROXMOX USER MANAGEMENT: ACCOUNTS AND PERMISSIONS [Електронний ресурс] – https://www.lectron.com/docs/faq%27/proxmox-security-and-access-control/proxmox-user-management-accounts-and-permissions/

47. Storage Replication [Електронний ресурс] – https://pve.proxmox.com/wiki/Storage_Replication

48. Backup and Restore [Електронний ресурс] – https://pve.proxmox.com/wiki/Backup_and_Restore

49. Firewall [Електронний ресурс] – https://pve.proxmox.com/wiki/Firewall

50. Proxmox 101 [Електронний ресурс] – https://medium.com/devops-dudes/proxmox-101-8204eb154cd5

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Магістерська кваліфікаційна робота на тему:

Дослідження сучасних технологій віртуалізації та реалізація їх за допомогою гіпервізора 1 типу "Proxmox"

Спеціальність 123 – Комп'ютерна інженерія

Керівник роботи
Черевик В'ячеслав Михайлович
к.н.т., доцент

Роботу підготував
Василенко Дмитро Євгенович
студент групи КСДМ-61

МЕТА, ОБ'ЄКТ, ПРЕДМЕТ ДОСЛІДЖЕННЯ

Мета роботи – провести глибокий аналіз сучасних технологій віртуалізації, зосереджуючись на гіпервізорі 1 типу "Proxmox".

Об'єкт дослідження – реалізація сучасних технологій віртуалізації за допомогою програмних продуктів WMware, VM VirtualBox та Proxmox.

Предмет дослідження – ESXi та vSphere від компанії WMware , Oracle VM VirtualBox та Proxmox.

АКТУАЛЬНІСТЬ ТЕМИ

АКТУАЛЬНІСТЬ ТЕМИ: Технології віртуалізації є вкрай актуальною темою у сучасному IT-середовищі. Зростаюча потреба в оптимізації використання ресурсів серверів та забезпеченні ефективного управління інфраструктурою створює важливі виклики, які можна вирішити шляхом використання технологій віртуалізації. Proxmox дозволяє розкрити переваги віртуалізації в контексті високої доступності та ефективного управління.

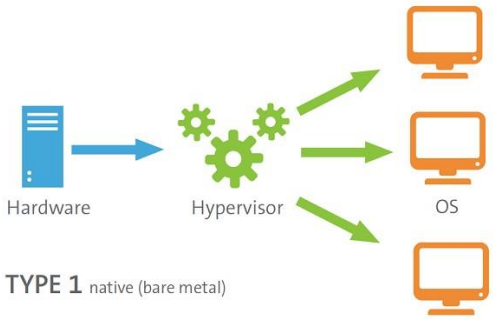
Технологія віртуалізації

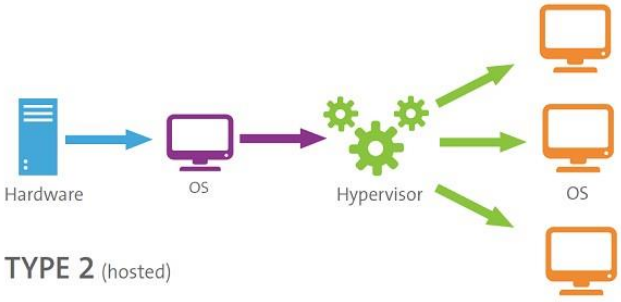
Віртуалізація – замість фізичної версії створюється імітоване чи віртуальне обчислювальне середовище. Віртуалізація дозволяє організаціям поділити один комп'ютер або сервер на кілька віртуальних машин.

Віртуальна машина

Віртуальна машина – модель обчислювальної машини, створеної шляхом віртуалізації обчислювальних ресурсів: процесора, оперативної пам'яті, пристроїв зберігання та вводу і виводу інформації.

Гіпервізор 1 та 2 типу

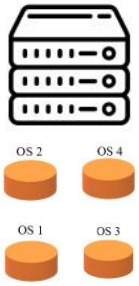




Віртуалізація серверної частини

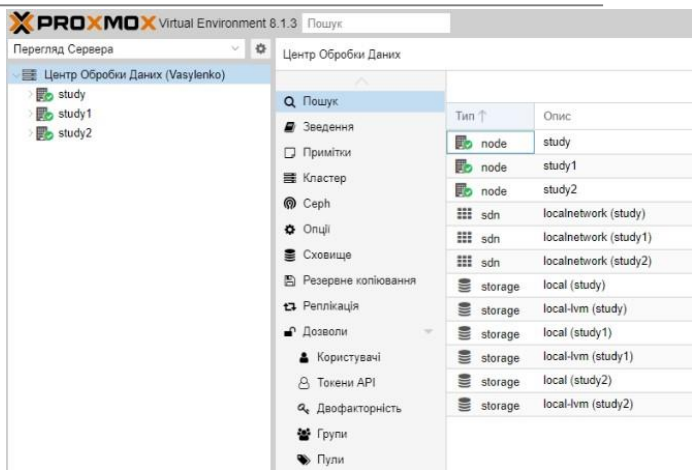
До віртуалізації		
Символ	Кількість	Опис
	1	Web Server
	1	Database server
	1	FTP server
	1	File server

Після віртуалізації		
Символ	Кількість	Опис
	1	Server



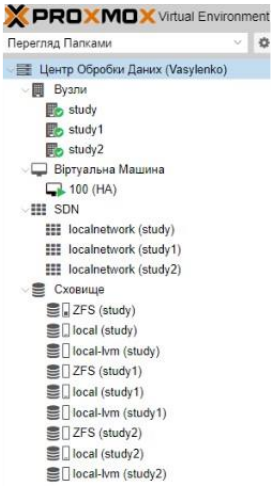
Proxmox

Proxmox є відкритим гіпервізором 1 типу, який поєднує в собі віртуалізацію контейнерів (LXC) та віртуальних машин (KVM) для оптимального управління ресурсами серверів. Його інтегрована платформа надає високий рівень управління та моніторингу, а також графічний інтерфейс для зручності користувача. Proxmox підтримує створення кластерів для забезпечення високої доступності та масштабованості системи. Завдяки своїй відкритій архітектурі та безкоштовній ліцензії, Proxmox став популярним рішенням для віртуалізації та управління серверами в області інформаційних технологій.



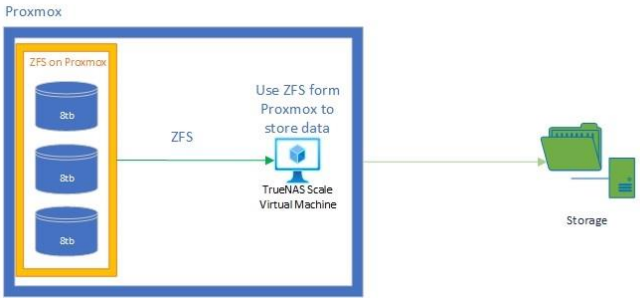
Кластеризація

Кластеризація в Proxmox дозволяє об'єднати кілька серверів в єдину систему для спільного управління та використання ресурсів. Забезпечує високу доступність та масштабованість віртуальних машин і контейнерів. Proxmox Cluster дозволяє автоматизувати відновлення роботи в разі виникнення проблем і забезпечує стійкість системи.

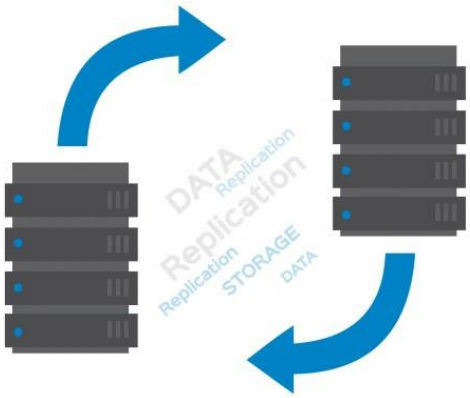


Спільне сховище ZFS

ZFS (Zettabyte File System) в Proxmox - це мощна система файлового сховища та управління даними, яка надає високий рівень надійності та ефективності. Забезпечує важливі функції, такі як відновлення даних, копіювання знімків, а також можливість розширення об'єму даних в реальному часі. Завдяки вбудованій підтримці ZFS в Proxmox, користувачі отримують високий рівень довіри та ефективного використання сховища для своїх віртуальних машин та контейнерів.



Реплікація



Реплікація в Proxmox є важливою стратегією для забезпечення надійності та високої доступності віртуальних машин. Цей механізм полягає в автоматичному створенні точних копій віртуальних машин та їх даних на інших серверах у кластері. В разі відмови або проблем на одному вузлі, реплікація дозволяє швидко включити резервний вузол, забезпечуючи неперервну роботу та мінімізуючи втрати даних.

High Availability Proxmox

High Availability (HA) в Proxmox - це інтегрована функціональність, яка дозволяє створювати кластери з кількох серверів для спільного управління та забезпечення високої доступності віртуальних машин та контейнерів. Завдяки автоматичному відновленню, HA переносить роботу з одного сервера на інший у разі відмови, уникнувши простою та забезпечуючи надійність системи.

Існує кілька вимог, яким має відповідати середовище перед налаштуванням HA Proxmox. Вони такі:

– Мінімально потрібно три вузли;

– Спільно використовуване сховище;

– Огородження (fencing).

ВИСНОВКИ

Досліджено сучасні технології віртуалізації та реалізовано їх задопомогою гіпервізора 1 типу "Proxmox", який розкриває значущі можливості для оптимізації IT-інфраструктури. "Proxmox" об'єднує в собі віртуалізацію контейнерів та віртуальних машин, надаючи широкий спектр інструментів для ефективного управління та моніторингу ресурсів. Дослідження розкриває високу доступність та надійність гіпервізора, його можливості розширення, а також зручний інтерфейс для користувачів.

Особливу увагу приділяється функціоналу кластеризації та High Availability "Proxmox", який забезпечує автоматичне переміщення віртуальних машин у разі відмови та ефективне розподілення завдань між серверами. Використання "Proxmox" дозволяє підвищити стійкість системи та забезпечити безперебійну роботу даже у випадку непередбачених обставин.

Це дослідження робить висновок, що "Proxmox" є перспективним рішенням для практичної реалізації віртуалізаційних потреб, забезпечуючи не лише технічну ефективність, але й високий рівень зручності в управлінні та масштабуванні IT-середовища.

Дякую за увагу!