УКРАЇНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ЗАЛІЗНИЧНОГО ТРАНСПОРТУ

ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КЕРУЮЧИХ СИСТЕМ
ТА ТЕХНОЛОГІЙ

Кафедра спеціалізованих комп'ютерних систем

МЕТОДИЧНІ ВКАЗІВКИ

до лабораторних робіт і самостійної роботи з дисципліни

«*ПАРАЛЕЛЬНІ ТА РОЗПОДІЛЕНІ ОБЧИСЛЕННЯ ТА CKLOUD-ТЕХНОЛОГІЇ*»

Харків 2023

Методичні вказівки розглянуто і рекомендовано до друку на засіданні кафедри спеціалізованих комп'ютерних систем 22 березня 2021 р., протокол № 10.

Описано основи паралельних і розподілених обчислень і CKLOUD-технологій.

Методичні вказівки призначено для здобувачів вищої освіти напряму 123 «Комп'ютерна інженерія», які вивчають курс «Паралельні та розподілені обчислення та CKLOUD-технології», денної та заочної форм навчання.

Укладачі:
проф. М. А. Мірошник,
асист. Ю. В. Савін,
викл. О. Л. Некрасов

Рецензент
доц. С. І. Доценко

ЗМІСТ

Вступ 4

Лабораторна робота 1. Ознайомлення з принципами побудови приватної, публічної та гібридної хмар для зберігання даних 6

Лабораторна робота 2. Ознайомлення з SAAS-, PAAS- та IAAS-схемами організації хмарних обчислень 29

Лабораторна робота 3. Ознайомлення з платформою хмарних обчислень GOOGLE APP ENGINE 76

Лабораторна робота 4. Ознайомлення з платформою хмарних обчислень MICROSOFT AZURE 117

Рекомендації щодо організації самостійної роботи 125

Список літератури 128

ВСТУП

Методичні вказівки призначені для проведення лабораторних робіт з дисципліни «Паралельні та розподілені обчислення та CKLOUD-технології» (ПРО CKLOUD Т), уведених до навчального процесу на підставі навчальних планів за вищевказаним напрямом. Метою викладання курсу є здобуття знань про принципи побудови та функціонування інформаційних систем за допомогою CKLOUD-технологій.

Методичні вказівки містять опис семи лабораторних робіт. Кожен розділ містить такі підрозділи: мета роботи, теоретична частина, вказівки з підготовки до виконання лабораторної роботи, зміст звіту, контрольні запитання.

За результатами вивчення дисципліни здобувачі повинні *знати* принципи організації сучасних CKLOUD-технологій і призначення, основні принципи функціонування систем на базі CKLOUD-технологій; різновиди CKLOUD-технологій; основні напрями розвитку в галузі CKLOUD-технологій і концептуальні перспективи їхнього розвитку; *уміти* розробляти архітектуру комп'ютерної системи з використанням CKLOUD-технологій; обирати CKLOUD-технологію відповідно до галузі застосувань; проєктувати комп'ютерні мережі у CKLOUD-середовищі; інтегрувати CKLOUD-технології в існуючі комп'ютерні системи; експлуатувати комп'ютерні системи, побудовані з використанням CKLOUD-технологій; створювати програмні додатки для CKLOUD-середовища.

Програмне забезпечення ЕОМ з дисципліни: VMware Workstation; Windows Virtual PC; Ceedo.

Оцінювання виконання роботи

Дисципліна «Паралельні та розподілені обчислення та CKLOUD-технології» закінчується розрахунково-графічною роботою та іспитом. При цьому виді контролю іспит вважається складеним успішно, якщо

протягом семестру здобувач отримав за всі контрольні заходи 60 балів за 100-бальною системою.

Оцінка «відмінно» (A, B (90 – 100)) – знати всі теми; оцінка «добре» (C (75 – 89)) – твердо знати мінімум знань і умінь; оцінка «задовільно» (D, E (60 – 74)) – мати мінімум знань і умінь. Також потрібно відпрацювати і захистити всі лабораторні роботи.

Лабораторна робота 1. Ознайомлення з принципами побудови приватної, публічної та гібридної хмар для зберігання даних

Мета роботи - ознайомитися з можливостями застосування віртуальних і віртуалізованих рішень від провідних вендорів хмарних послуг.

1.1 Вказівки з підготовки до виконання лабораторної роботи

1.1.1 Google CKLOUD

Основний напрям розвитку віртуалізованих рішень корпорації Google лежить у площині розбудови власної платформи Google CKLOUD Platform і Google Apps.

Google Apps – це середовище, яке надає такі засоби спільної роботи: вже популярний поштовий сервіс GMail, клієнт обміну миттєвими повідомленнями Google Hangouts, календар Google Calendar, засоби для роботи з документами та електронними таблицями Google Docs&Spreadsheets, редактор сторінок від Google для швидкого створення і опублікування потрібної інформації [1–7].

Календар. Дуже зручний засіб планування особистого робочого часу, який в умовах глобальної інтеграції дає змогу планувати не тільки свій робочий час, а і враховувати робочий час і завдання колег. Основні можливості календаря в Google Apps: створення подій, для кожної з яких можна визначити назву події, час і тривалість, склад учасників з перевіркою їхньої зайнятості під час події, установлення нагадувань про події, перегляд чужих календарів, робота з календарем на мобільних пристроях, управління доступом до календарів тощо.

Користувач домену може додати календар конкретного співробітника, використовуючи спеціальну форму, яка дає змогу шукати календарі, використовуючи як ключові слова, так і адресу електронної пошти співробітника. Отже, можна завжди мати актуальну інформацію про завдання та роботу певного співробітника (очевидно, тільки настільки, наскільки це дає змогу визначити його календар).

Робота з документами. Підтримуються всі популярні формати документів: Word, Excel, OpenOffice. На сьогодні доступна можливість роботи з такими типами файлів: документи Word і Excel, OpenOffice, RTF, HTML і текстові документи. Такий набір підтримуваних форматів забезпечує придатність сервісу для широкого кола користувачів. Результати роботи можуть бути як збережені на локальний комп'ютер, так і залишені на зберігання на сервері.

Отже, для повноцінної роботи з документами досить просто мати доступ до сервісу Google Apps з будь-якої точки світу, будь-якого комп'ютера.

Порівняння можливостей самих редакторів Google Docs&Spreadsheets з Word і Excel або OpenOffice є темою для окремої дискусії. Практика використання показує, що функцій вистачає для того, щоб підготувати нормальний документ, який містить елементи оформлення, що зазвичай використовуються: списки, форматування і різні стилі, таблиці, зображення і гіперпосилання тощо.

Google Docs&Spreadsheets надає широкі можливості для спільної роботи над документами. Для того щоб зробити роботу з документами якомога зручнішою і продуктивною для групи співробітників, використовуйте такі опції:

– управління версіями документа. Проміжні версії документа створюються системою автоматично досить часто і, крім того, кожен раз, коли користувач зберігає документ. Доступна функція порівняння двох обраних версій, що дає змогу легко відстежувати зміни, внесені черговим редагуванням документа;

– управління доступом до документа. Можна запрошувати користувачів системи до спільної роботи з документом, вказуючи, які права даються користувачеві: тільки перегляд документа або редагування. Додаткові можливості дають можливості користувачеві, якому було запропоновано працювати з документом, у свою чергу запрошувати інших користувачів. Люди, які одночасно працюють з документом, можуть

влаштовувати чат для обговорення змін у документі, що будуть видимі всім учасникам обговорення. Доступна можливість публікації документа з постійною адресою, що дає можливість будь-якому співробітникові домену отримувати доступ до документа (після того як усі зміни зроблені і схвалені, документ публікується для загального ознайомлення). Крім того, існує можливість архівування документів, які вже доведені до їхнього фінального стану, але ще можуть знадобитися.

Спільна робота з документами в Google Apps організована на високому рівні, достатньому для того, щоб працювати з документами і швидко та ефективно отримувати доступ до загальної інформації. А призначені для користувача функції редагування вмісту документів лише в окремих аспектах поступаються звичним Word, Excel і OpenOffice.

Стартова сторінка і редактор сторінок. Стартова сторінка – перше, що побачать користувачі після входу в Google Apps. Стартова сторінка – це місце, призначене для того, щоб бути першим, що побачать користувачі після входу в систему Google Apps. Ця сторінка схожа з персональною сторінкою Google (www.igoogle.com), так само вона призначена бути першою сторінкою, з якою користувач стикається, починаючи роботу. На ній можуть бути розташовані гаджети від Google і сторонніх розробників, а також інформація, необхідна для початку роботи співробітників. Неповний список корисних речей, які можна розмістити на стартовій сторінці Google Apps: гаджети для попереднього перегляду особистих поштових скриньок і подій у календарі, пошук, у тому числі можна і спеціальний Google Custom Search, який дає змогу шукати тільки те, що реально необхідно, перегляд RSS, закладки, Google Notebook і ще багато корисних елементів.

Редактор веб-сторінок дає змогу легко і швидко створювати власні сторінки, які потім зручно публікувати за допомогою сервісу Google. Для створення можна використовувати велику кількість уже готових дизайнів (розташування елементів на сторінці, колірна гама тощо) і досить зручний редактор, який практично не потребує від користувача знання HTML, CSS

та інших мов. Публікація нової сторінки відбувається миттєво, що суттєво скорочує час, який витрачається на те, щоб донести інформацію до інших користувачів або клієнтів.

Інтеграція в Google Apps досягла того, що документи, які надійшли поштою, одразу з поштової скриньки можна відкрити для роботи в Google Docs & Spreadsheets. Перемикання між різними сервісами здійснюється простим натисканням мишкою. Втім, нема проблем у тому, щоб тримати їх усі відкритими і готовими для роботи в різних вікнах браузера.

Отже, **Google Apps** створює собою середовище, зручне не тільки для спільної роботи кількох людей, які перебувають у різних місцях, але й організації спільної роботи великої кількості співробітників, які можуть бути розкидані по різних країнах. З огляду на те, що компанія Google досить часто надає API до своїх продуктів, тим самим даючи можливість створювати власні додатки, які ще більше інтегрують додатки Google в конкретне середовище роботи, можна сказати, що Google Apps може стати дуже зручним середовищем для спільної роботи. Крім того, компанія Google постійно веде роботу з поліпшення існуючих сервісів і додавання нових. У найближчих планах додавання сервісу для міграції з різних поштових клієнтів на GMail і можливість створювати і редагувати презентації, тобто сервіс, аналогічний програмі PowerPoint.

Google App Engine дає змогу виконувати веб-додатки в інфраструктурі Google. Додатки App Engine легко створювати, підтримувати і вдосконалювати зі збільшенням трафіка і сховища даних. При роботі з App Engine не потрібно підтримувати сервер: просто завантажити свій додаток, і користувачі зможуть працювати з ним. Додаток можна опублікувати у власному домені (наприклад http://www.example.com/) за допомогою служб Google або скористатися безкоштовним ім'ям у домені appspot.com. Додаток можна зробити доступним для всіх або надати доступ тільки учасникам певного колективу.

Google App Engine підтримує програми, написані кількома мовами програмування. Завдяки середовищу виконання Java App Engine можна створювати додатки за допомогою стандартних технологій Java, у тому числі JVM, сервлетів Java і мови програмування Java або іншої мови, що використовує інтерпретатор або компілятор на JVM, наприклад JavaScript або Ruby. Крім того, App Engine надає спеціальне середовище виконання Python, яке включає швидкий інтерпретатор і стандартну бібліотеку Python. Середовища виконання Java і Python розроблені спеціально для того, щоб програми могли швидко і безпечно виконуватися без взаємодії з іншими додатками в системі.

З App Engine потрібно платити тільки за те, що використано. Не потрібно платити за установлення або вносити періодичні платежі. Оплата за використані додатком ресурси, такі як об’єм зберігання і трафік, вимірювані в гігабайтах, стягується за обґрунтованими ставками. Можна управляти максимальною кількістю ресурсів, які додаток може використовувати, що дасть можливість завжди залишатися в межах бюджету.

Почати використовувати App Engine можна безкоштовно. Не доведеться платити за програми, які використовують менше 500 Мбайт сховища, а також ресурси ЦП і трафік, достатні для ефективного застосування, обслуговують до п'яти мільйонів переглядів сторінок на місяць. Із виключенням оплати для додатка ці обмеження підвищуються, але все одно будуть оплачуватися тільки ресурси, використані понад безкоштовні рівні.

Середовище програм. **Google App Engine** дає змогу легко створювати додатки, що надійно працюють навіть при великому навантаженні і з великими об'ємами даних. App Engine включає такі функції [1–7]:

– динамічну роботу в Інтернеті з повною підтримкою основних вебтехнологій;

– постійне сховище з запитами, сортуванням і транзакціями;

– автоматичне масштабування і регулювання навантаження;

– API для аутентифікації користувачів і відправлення електронної пошти за допомогою облікового запису Google;

– повнофункціональне локальне середовище розроблення, що імітує Google App Engine на комп'ютері;

– заплановані завдання для відстеження подій у певний час або через регулярні інтервали.

Додаток може виконуватися в одному з двох середовищ виконання: Java і Python. Кожне середовище надає стандартні протоколи і основні технології для розроблення вебдодатків.

Додатки працюють у безпечному середовищі, що забезпечує обмежений доступ до операційної системи, яка додається. Обмеження дають змогу App Engine поширювати вебзапити для додатка на кілька серверів, а також запускати і зупиняти сервери залежно від трафіка. Тестове середовище ізолює додаток у власному безпечному і надійному середовищі, незалежному від обладнання, операційної системи і фізичного розташування вебсервера.

Нижче наведені приклади обмежень надійного тестового середовища. Додаток може отримувати доступ до інших комп'ютерів в Інтернеті тільки через надані API, службу отримання даних по URL і службу електронної пошти. Інші комп'ютери можуть підключатися до додатка тільки шляхом HTTP-запитів (або HTTPS-запитів) через загальні порти.

Додаток не може записувати відео у файлову систему; може зчитувати файли, але тільки ті, що були завантажені разом з кодом програми. Додаток має використовувати сховище даних App Engine, кеш пам'яті й інші служби для всіх даних, що зберігаються між запитами.

Код додатка виконується тільки у відповідь на вебзапит або завдання Cron і в будь-якому випадку має повертати ці відповіді протягом 30 с. Обробник запитів не може створити підпроцес або виконати код після надсилання відповіді.

Для середовища виконання Java можна розробити програму за допомогою стандартних інструментів веброзробки Java і стандартних API. Додаток взаємодіє із середовищем за допомогою стандарту Java Servlet і може використовувати стандартні технології вебдодатка, такі як сторінки JavaServer Pages (JSP).

Середовище виконання Java використовує Java 6. SDK Java App Engine підтримує розроблення додатків за допомогою Java 5 і 6. Середовище включає платформу Java SE Runtime Environment (JRE) 6 і бібліотеки. Обмеження на тестове середовіще реалізовані у JVM. Додаток може використовувати байтовий код JVM або бібліотеки в межах обмежень тестового середовища. Наприклад, при спробі байтового коду відкрити сокет або записати файл виникне вимкнення середовища виконання.

Доступ до більшості служб App Engine можна отримати через стандартні API Java. Для сховища даних App Engine SDK Java містить реалізації інтерфейсів об'єктів даних Java (JDO) і Java Persistence API (JPA). Щоб відправляти повідомлення електронною поштою за допомогою служби Mail App Engine, можна використовувати API JavaMail. У API HTTP java.net є доступ до служби отримання даних по URL App Engine. Крім того, для своїх служб App Engine включає низькорівневі API, які реалізують додаткові адаптери і дають змогу використовувати служби безпосередньо з програми. Ознайомтеся з документацією щодо API-сховища даних, кеша пам'яті, отримання даних по URL, пошти, зображень і акаунтів Google.

Зазвичай, щоб розробити вебдодатки для JVM, Java-розробники використовують мови програмування Java і API. Використовуючи сумісні з JVM компілятори та інтерпретатори, можна розробляти вебдодатки іншими мовами, такими як JavaScript, Ruby і Scala.

Завдяки середовищу виконання Python App Engine можна створювати додатки за допомогою мови програмування Python і виконувати їх за допомогою оптимізованого інтерпретатора Python. App

Engine включає різноманітні API і інструменти для розроблення вебдодатків Python, у тому числі API-моделювання збагачених даних, просту у використанні інфраструктуру вебдодатків і інструменти для управління і доступу до даних додатка. Для розроблення вебдодатків Python можна скористатися перевагами широкого набору бібліотек і інфраструктур, наприклад Django.

Середовище виконання Python використовує Python версії 2.5.2. Для майбутнього випуску розглядається можливість підтримки Python 3.

Середовище Python містить стандартну бібліотеку Python. Природно, не всі функції бібліотеки можна використовувати в тестовому середовищі. Наприклад, при виклику методу, який відкриває сокет або записує у файл, виникне відключення. Для зручності відключені кілька модулів стандартної бібліотеки, ключові функції яких не підтримуються середовищем виконання. Виконання коду, що імпортує їх, призводить до помилки.

Код додатка, створений для середовища Python, має бути написаний виключно на Python. Розширення, написані мовою C, не підтримуються.

Середовище Python надає потужні API Python для служб сховища даних, облікового запису Google, отримання URL та електронної пошти. App Engine також надає просту інфраструктуру вебдодатків Python під назвою webapp, яка полегшує створення додатків.

Разом з додатком можна завантажувати сторонні бібліотеки, але вони мають бути реалізовані на чистому Python і не потребувати підтримки модулів стандартної бібліотеки.

App Engine надає потужну службу розподіленого зберігання даних, що включає механізм запитів і транзакції. Розширення розподіленої бази даних за рахунок даних аналогічно розширенню розподіленого вебсервера за рахунок трафіка.

Сховище даних App Engine не схоже на звичайну реляційну базу даних. Об'єкти даних, або записи, мають вигляд і володіють набором властивостей. За допомогою запитів можна отримувати записи певного

видгляду, відфільтровані і відсортовані за значеннями властивостей. Значення властивостей можуть бути будь-якими з підтримуваних типів значень властивостей.

Для об'єктів сховища даних не потрібна схема. Структура об'єктів даних визначається в коді програми. Інтерфейси JDO / JPA Java і сховища даних Python включають функції для застосування структури в додатку. Додаток може отримати прямий доступ до сховища даних, щоб реалізувати потрібну частину структури.

Сховище даних узгоджено і використовує оптимістичне управління паралельними транзакціями. Оновлення запису відбувається в транзакції, виконуваній повторно певну кількість разів, якщо інші процеси одночасно намагаються оновити той самий запис. Додаток може виконувати кілька операцій зі сховищем даних в одній транзакції. Ці всі операції будуть або успішними, або невдалими, що забезпечує цілісність даних.

Сховище даних реалізує транзакції у своїй розподіленій мережі за допомогою груп записів. Транзакція чинить дії над записами в одній групі. Записи кожної з груп зберігаються разом для ефективного виконання транзакцій. При створенні записів додаток може приєднувати їх до груп.

App Engine підтримує інтеграцію додатків з акаунтами Google для аутентифікації користувачів. Ваша програма може дозволити користувачеві увійти у свій аккаунт Google і отримати доступ до адреси електронної пошти і псевдоніма, пов'язаних з членством. Використання облікових записів Google дає користувачеві можливість швидше почати використовувати ваш додаток, оскільки йому не доведеться створювати новий акаунт. Це також знімає з вас необхідність реалізовувати систему акаунтів користувачів тільки для свого застосування. Якщо додаток працює в службах Google, він може використовувати ті самі функції для учасників вашої організації і акаунтів служб Google. З додатком API-користувачів можна також сказати, чи поточний користувач зареєстрований адміністратором програми. Це спрощує реалізацію адміністративних зон сайту. App Engine надає набір служб, що дають

змогу виконувати рядові операції при управлінні додатком. Для доступу до цих служб надані API. Додатки можуть отримувати доступ до ресурсів в Інтернеті, наприклад до вебслужб або інших даних, за допомогою служби отримання URL App Engine. Служба отримання даних по URL забезпечує отримання вебресурсів за допомогою тієї самої високошвидкісної інфраструктури Google, яка отримує вебсторінки для багатьох інших продуктів Google.

Електронна пошта. Додатки можуть відправляти повідомлення електронної пошти за допомогою поштової служби App Engine. Для відправлення електронних повідомлень ця служба використовує інфраструктуру Google [1–7].

Служба **Memcache** надає вашому додатку високопродуктивний кеш пам'яті, що використовує структуру ключ-значення, до якого можуть отримувати доступ кілька примірників додатків. Кеш пам'яті стане в нагоді для даних, які потребують постійного зберігання та функції роботи з транзакціями, що надає сховище даних, наприклад для часових даних або даних, що копіюються зі сховища в кеш для прискорення доступу [1–7].

Робота з зображеннями. Служба зображень дає змогу додатку працювати з зображеннями. За допомогою цього API можна змінювати розмір, обрізати, повертати і відображувати зображення у форматах JPEG і PNG.

Заплановані завдання. Служба Cron дає змогу планувати завдання для виконання через певні інтервали. Детальніше про неї можна дізнатися в документації зі служби Cron Python і Java.

Процес розроблення. Інструментарій розроблення App Engine (SDK) для Java і Python включає додаток на вебсервері, який імітує служби App Engine на локальному комп'ютері. Кожен SDK включає всі API і бібліотеки, доступні в App Engine. Крім того, вебсервер імітує безпечне тестове середовище, що включає перевірку на доступ до системних ресурсів, заборонену в App Engine [1–7].

Кожен SDK також включає інструмент для додавання додатка в App Engine. Після створення коду програми, статичних файлів і файлів конфігурації запустіть цей інструмент, щоб завантажити дані. Інструмент запросить адресу електронної пошти і пароль вашого облікового запису Google.

При створенні нового випуску програми, який вже працює в App Engine, ви зможете завантажити його як нову версію. Стара версія буде працювати для користувачів доти, поки ви не перейдете на нову. Ви можете тестувати нову версію в App Engine, поки працює стара.

SDK Java виконується на будь-якій платформі з Java 5 або Java 6. SDK доступний у вигляді ZIP-файла. При використанні середовища розроблення Eclipse, щоб створити, перевірити і додати додатки App Engine, можна використовувати плагін Google для Eclipse. SDK також містить інструменти, які працюють з командного рядка, що дають змогу запускати сервер розроблення і додавати додатки.

SDK Python реалізований на чистому Python і виконується на будь-якій платформі з Python 2.5, у тому числі Windows, Mac OS X і Linux. SDK доступний у вигляді Zip-файлу, а для Windows і Mac OS X доступні програми установлення. Консоль адміністрування – це вебінтерфейс для управління додатками, що працюють в App Engine. Її можна використовувати для створення нових додатків, налаштування доменних імен, зміни робочої версії додатка, вивчення доступу і журналів помилок і перегляду сховища даних програми.

1.1.2 Microsoft Azure

«**Microsoft Azure**» – це хмарна платформа та інфраструктура корпорації Microsoft, призначена для розробників застосунків хмарних обчислень і покликана спростити процес створення онлайнових додатків [1–7].

Корпорація Microsoft презентувала платформу Windows Azure 27 жовтня 2008 р. Windows Azure дає змогу створювати додатки як за

допомогою Microsoft.NET Framework і Visual Studio, так і інших інструментів. Операційна система працює на серверах Microsoft, доступ до неї можна отримати за протоколами HTTP, Representational State Transfer (REST), WS–* і Atom Publishing Protocol (AtomPub).

Платформа Azure Services Platform включає п'ять основних компонентів. Це сама операційна система Windows Azure, що управляє дисковим простором, додатками і мережами, і Microsoft SQL Services для роботи з базами даних. Також у платформу входять Microsoft.NET Services, Live Services, і бізнес-компонент, що включає Microsoft SharePoint Services і Microsoft Dynamics CRM Services.

Концепція хмарних обчислень – це використання обчислювальних потужностей, дискового простору і каналів зв'язку «обчислювальної хмари» для виконання трудомістких завдань. Навантаження між комп'ютерами, що входять у цю хмару, розподіляється автоматично. Більшість хмарних застосунків працюють у браузері. На момент оголошення розробка Windows Azure, за словами представників Microsoft, все ще перебуває на ранній стадії. Зараз на Azure можуть запускатися тільки застосунки на основі .NET Framework. Очікується, що можливості системи будуть значно розширені. Крім того, усі сервіси для Windows Azure мають бути побудовані на базі наперед заданих шаблонів, хоча Microsoft планує незабаром значно збільшити їхню кількість, а також дати можливість створювати застосунки, які не вписуються в шаблони.

За оцінками аналітиків, анонс Azure є «захисним маневром», щоб нинішні клієнти Microsoft, які користуються її серверами й іншими продуктами, не переходили на хмарні рішення від таких компаній, як Amazon чи IBM, що вже надають низку сервісів для зберігання даних і обчислювальні потужності у хмарі.

У грудні 2009 р. Microsoft об'єднав існуючі підрозділи, що займалися Windows Server та Azure, в один Server & CKLOUD Division (SCD).

Платформа виготовлена з групи із трьох технологій, що забезпечують спеціалізований набір можливостей для розробників. Більш

того, платформу Windows Azure можна використовувати в додатках, що працюють локально на комп'ютерах користувачів або у хмарі.

Платформа Windows Azure [1–7] – надає середовище виконання для додатків, заснованих на операційних системах і Windows Server, а також місця для зберігання даних. Система працює на віртуальних машинах за допомогою аналогічної технології Hyper-V$ обчислення – відповідає за обчислення розміщення додатків; зберігання – відповідає за зберігання даних у хмарі; SQL Azure – дає можливість використовувати реляційну базу даних для запуску в хмарі.

Платформа Windows Azure AppFabric – компонент, який забезпечує додаткову функціональність у вигляді послуг.

Технологіями, які можуть бути використані для створення додатків Windows Azure Platform, є всі технології, які можуть бути запущені на платформі Windows. Разом із технологією .net (посилання технології для Windows Azure хмари) можна використовувати такі технології: Java, PHP, C / C ++ або Python.

Важливим компонентом є SDK – емулятор хмари. Основні відмінності: емулятор дає можливість підключатися до управління на його налагодження під час виконання. Правда, платформа Windows Azure не дасть змоги. Примірник працює на емуляторі, він має доступ до бібліотек, зареєстрованих у місцевому глобальному кеші збірок (GAC), для реєстрації та конфігурації комп'ютера. Ці послуги доступні не на фактичній платформі. Емулятор дає змогу запам'ятовувати діагностичну інформацію на консолі або використовувати Windows Azure Diagnositics. Платформа Windows Azure запису інформації можлива тільки за допомогою Windows Azure Diagnostics; уся інформація зберігається в спеціальній таблиці у Windows Azure Storage. Усі екземпляри, що працюють на емуляторі, мають привілеї адміністратора і ті, які працюють на платформі, є силою стандартного користувача Windows. Емулятор не повністю відображує поведінку балансування навантаження, використовувану на Windows Azure Platform.

1.1.3 Amazon Web Services

Amazon Web Services є дочірньою компанією Amazon.com, що надає платформу хмарних обчислень в оренду приватним особам, компаніям та урядам на основі платної підписки. Існує і безкоштовна підписка, доступна протягом перших 12 місяців. Технологія дає можливість абонентам мати у своєму розпорядженні повноцінний віртуальний кластер комп'ютерів, який завжди доступний через Інтернет. Віртуальні комп'ютери AWS мають більшість атрибутів реального комп'ютера, включаючи апаратні пристрої (процесор, відеокарту, локальну та оперативну пам'ять, жорсткий диск або SSD-накопичувач); операційну систему (ОС) на вибір; мережу; попередньо встановлені прикладні програми, такі як вебсервер, база даних, CRM тощо. Кожна система AWS також віртуалізує консольне введення / виведення (клавіатура, дисплей і миша), що дає змогу користувачам AWS підключитися до своєї системи AWS за допомогою браузера. Браузер виступає як вікно у віртуальний комп'ютер, що дає змогу користувачу входити в систему, налаштовувати та використовувати свої віртуальні системи так само, як справжній, фізичний комп'ютер. Це дає можливість їм налаштувати систему так, щоб надавати Інтернет-орієнтовані сервіси та послуги своїм клієнтам [1–7].

Технологія AWS базується на серверних кластерах (фермах), розташованих по всьому світі. Плата за користування базується на комбінації використання апаратних засобів / ОС / програмного забезпечення (ПЗ) / мережевих функцій, вибраних користувачем, а також вимог до доступності (availability), надлишковості (redundancy), безпеки та додаткових параметрів. Виходячи з того, що користувач потребує і оплачує, він може зарезервувати один віртуальний комп'ютер (VM), кластер віртуальних комп'ютерів (VM Cluster), фізичний (реальний) комп'ютер (Server), призначений для його виняткового використання, або навіть кластер фізичних комп'ютерів (Server Cluster). Компанія Amazon зобов'язується управляти та оновлювати програмне та апаратне забезпечення для дотримання необхідних стандартів безпеки. AWS працює

в багатьох географічних регіонах, у тому числі Канаді, Німеччині, Ірландії, Сінгапурі, Токіо, Сіднеї, Пекіні, Лондоні і т. д.

У 2016 р. AWS надавав більш ніж 70 сервісів, що охоплюють широкий спектр, включаючи обчислення та зберігання даних, їхню передачу по мережі, аналітику, мобільні додатки, інструменти для розробників тощо. Найпопулярнішими з них є Amazon Elastic Compute CKLOUD (EC2) і Amazon Simple Storage Service (S3). Більшість служб не надаються безпосередньо кінцевим користувачам, але замість цього пропонуються функціональні можливості через API, які розробники можуть використовувати у своїх програмах. Пропозиції Amazon Web Services доступні через HTTP, використовуючи архітектурний стиль REST та протокол SOAP. Amazon рекламує AWS як спосіб отримання обчислювальної потужності, що масштабується швидше та дешевше, ніж побудова власного фізичного серверного кластера. Усі послуги оплачуються залежно від використання, однак кожна служба вимірює використання своїм методом.

Список рішень:

– **галузь обчислень**:

а) Amazon Elastic Compute CKLOUD (EC2) – це сервіс IaaS, що надає в користування віртуальні сервери, які контролюються API, засновані на гіпервізорі Xen. Еквівалентні віддалені сервіси включають Microsoft Azure, Google Compute Engine та Rackspace; рішення, які встановлюються на локальні сервери OpenStack або Eucalyptus [1–7];

б) Amazon Elastic Beanstalk – надає сервіс PaaS для розміщення хостингу програм. Еквівалентні сервіси: Google App Engine, Heroku та OpenShift для локального використання;

в) Amazon Lambda (AWS Lambda) – запускає код у відповідь на внутрішні або зовнішні події, такі як http запит, відкрито надаючи необхідні ресурси. Lambda глибоко інтегрована з AWS, але подібні сервіси, такі як Google CKLOUD Functions та відкриті рішення типу OpenWhisk, набирають популярності;

– галузь мережевих технологій:

а) Amazon Route 53 – надає сервіс Managed DNS, що масштабується, який у свою чергу надає сервіс перетворення імені хоста на IP-адресу;

б) Amazon Virtual Private CKLOUD (VPC) – створює логічний, ізольований набір ресурсів AWS, які можуть бути об'єднані за допомогою VPN. Рішення конкурентів – це OpenStack або HPE Helion Eucalyptus, що використовується в поєднанні з програмним забезпеченням PaaS;

в) AWS Direct Connect – надає виділені мережеві підключення до дата центрів AWS [1–7];

г) Amazon Elastic Load Balancing (ELB) – автоматично розподіляє вхідний трафік між багатьма інстансами Amazon EC2;

д) AWS Elastic Network Adapter (ENA) – надає мережевий канал шириною 20Gbit/s до інстансів Amazon EC2;

– **галузь доставки контету**: Amazon CloudFront – це мережа доставки контенту (CDN) до обладнання, що фізично перебуває неподалік від запиту;

– **галузь побудови контакт центру**: Amazon Connect – центр самообслуговування. Сервіс контакт-центру базується на хмарному рішенні. Він дає можливість багатьом бізнесам надавати кращу підтримку для клієнтів за нижчою ціною. Amazon Connect базується на тій самій технології, що й Amazon customer service;

– **галузь зберігання данних та доставка контенту**:

а) Amazon Simple Storage Service (S3) – забезпечує зберігання даних типу об'єкт (object storage), масштабується та доступний через інтерфейс Web Service. Застосовується для створення резервних копій / архівування, зберігання файлів (включаючи медіа) і хостингу, хостингу статичних вебсторінок, даних програми тощо [1–7];

б) Amazon Glacier – використовується для довготермінового зберігання даних (порівняно з S3). Гарантує високу надлишковість (redundancy) і доступність (availability), проте має повільний доступ до даних. Призначений для архівування даних;

в) AWS Storage Gateway - віртуальний пристрій зберігання блоків iSCSI з підтримкою хмарних резервних копій;

г) Amazon Elastic Block Store (EBS) – забезпечує постійні об’єми зберігання на рівні блоків для EC2;

д) AWS Import/Export – прискорює переміщення великих об'ємів даних у / із AWS, використовуючи портативні пристрої зберігання для транспортування;

е) Amazon Elastic File System (EFS) – це служба зберігання файлів (файлова система) для інстансів Amazon Elastic Compute CKLOUD (Amazon EC2);

– **галузь баз даних**:

а) Amazon DynamoDB – надає масштабовану онлайн-базу даних NoSQL з низьким часом відклику. Це забезпечується тим, що база даних біжить на SSD;

б) Amazon ElastiCache – забезпечує кешування вебпрограм, що знаходяться в пам'яті. Аналогічні сервіси: Memcached та Redis [1–7];

в) Amazon Relational Database Service (RDS) – надає масштабовані сервери баз даних з підтримкою MySQL, Oracle, SQL Server та PostgreSQL;

г) Amazon Redshift – забезпечує зберігання даних у масштабі петабайтів за допомогою накопичувача на основі стовпців і численних вузлів;

д) Amazon SimpleDB – дає змогу розробникам запускати запити щодо структурованих даних. Він працює в поєднанні з EC2 і S3;

е) AWS Data Pipeline – забезпечує надійний сервіс для передачі даних між різними службами обчислення та зберігання AWS (наприклад Amazon S3, Amazon RDS, Amazon DynamoDB, Amazon EMR). Іншими словами, ця служба – це просто система управління робочим навантаженням, яка надає API для управління та моніторингу робочих навантажень у хмарних додатках;

ж) Amazon Aurora – забезпечує MySQL-сумісний реляційний двигун бази даних, створений спеціально для інфраструктури AWS, що надає більшу швидкість і зменшує витрати порівняно з великими базами даних;

– **галузь розгортання додатків**:

а) AWS CKLOUDFormation – забезпечує декларативну інфраструктуру на основі моделі «Інфраструктура як код» для налаштування AWS;

б) AWS Elastic Beanstalk – забезпечує розгортання та управління додатками в хмарі;

в) AWS OpsWorks – дає можливість налаштовувати сервіси EC2, використовуючи Chef;

г) AWS CodeDeploy – дає можливість автоматизувати розгортання коду на інстансах EC2 [1–7];

– **галузь менеджменту**:

а) Amazon Identity and Access Management (IAM) – це сервіс, що дає змогу безпечно контролювати доступ до сервісів AWS і ресурсів для ваших користувачів. Використовуючи IAM, ви можете створювати та управляти користувачами та групами, надавати чи забороняти доступ до ресурсів і сервісів, змінюючи права доступу [1–7];

б) AWS Directory Service – це служба, яка дає можливість підключення до ресурсів AWS з увімкнутою локальною службою Microsoft Active Directory або для створення нового окремого каталогу в AWS Cloud;

в) Amazon CloudWatch – забезпечує моніторинг хмарних ресурсів і програм AWS, починаючи з EC2;

г) AWS Management Console (AWS Console) – візуальний вебінтерфейс для управління та моніторингу інфраструктури Amazon, включаючи (але не обмежуючись) EC2, EBS, S3, SQS, Amazon Elastic MapReduce and Amazon CKLOUDFront. Мобільна програма для Android підтримує деякі функції управління з консолі;

д) Amazon CloudHSM – ця служба допомагає відповідати корпоративним, договірним і нормативним вимогам щодо забезпечення безпеки даних, використовуючи спеціальні пристрої для захисту обладнання (HSM) у хмарі AWS;

е) AWS Key Management Service (KMS) – це керована служба для створення та управління ключами шифрування;

ж) Amazon EC2 Container Service (ECS) – це високомасштабна та швидка служба управління Docker контейнерами;

– **галузь служби адміністрування**:

а) Amazon API Gateway – служба для публікації, підтримки та захисту API вебсервісів [1–7];

б) Amazon CloudSearch – забезпечує базовий текстовий пошук та індексування текстового контенту;

в) Amazon DevPay – білінг-система та система управління обліковими записами для програм, розроблених поверх вебсервісів Amazon (Amazon Web Services);

г) Amazon Elastic Transcoder (ETS) – забезпечує перекодування відео, розміщене на S3, перш за все як спосіб перетворення файлів у версії для мобільних пристроїв;

д) Amazon Simple Email Service (SES) – надає групове і транзакційне надсилання електронної пошти;

е) Amazon Simple Queue Service (SQS) – забезпечує розміщення черги повідомлень для вебпрограм;

ж) Amazon Simple Notification Service (SNS) – забезпечує розміщення мультипротокольних push повідомлень для додатків;

и) Amazon Simple Workflow (SWF) – це робочий процес для створення масштабованих і гнучких програм;

к) Amazon Cognito – це служба ідентифікації користувачів і синхронізації даних, яка безпечно управляє та синхронізує дані додатків користувачів на їхніх мобільних пристроях;

л) Amazon AppStream 2.0 – служба зі швидким мережевим відкликом, яка транслює ресурсомісткі додатки та ігри з хмари використовуючи технології NICE DVC;

– галузь аналітики:

а) Amazon Athena – інтерактивна служба запитів, яка полегшує аналіз даних в Amazon S3, використовуючи стандартний SQL. Athena безсерверна, тому для управління не існує інфраструктури, і ви платите за запити, які запускаєте;

б) Amazon Elastic MapReduce (EMR) – забезпечує PaaS сервіс, який надає фреймворк Hadoop для запуску запитів MapReduce. Працює на інфраструктурі EC2 та Amazon S3 [1–7];

в) Amazon Machine Learning – допомагає розробникам усіх рівнів кваліфікації використовувати технологію машинного навчання;

г) Amazon Kinesis – хмарний сервіс для обробки великої кількості розподілених потоків даних у режимі реального часу. Сервіс транслює дані в режимі реального часу з можливістю обробки тисяч потоків даних за секунду. Дає змогу розробникам витягувати будь-яку кількість даних з будь-якої кількості джерел, збільшуючи або зменшуючи кількість джерел за необхідності. Він має деяку схожість за функціоналом з Apache Kafka;

д) Amazon Elasticsearch Service – забезпечує повністю керовані послуги Elasticsearch та Kibana;

е) Amazon QuickSight – інструмент бізнес-аналізу, аналітики та візуалізації. Він надає спеціальні послуги шляхом підключення до джерел даних AWS або стороніх джерел;

– інші галузі:

а) Amazon Marketplace Web Service (MWS) – інтегрований веб API, який допомагає продавцям на Amazon програмно обмінюватися списками, замовленнями, платежами, звітами тощо. Інтеграція даних з Amazon забезпечує високий рівень автоматизації продаж, що може допомогти продавцям розвивати свій бізнес. Використовуючи Amazon MWS, продавці можуть підвищити ефективність продажів, знизити вимоги до праці та скоротити час відповіді клієнтів. Amazon MWS – безкоштовний сервіс, але для його використання вам необхідно мати обліковий запис продавця Amazon MWS і зареєструватися для його використання [1–7];

б) Amazon Fulfillment Web Service – надає продавцям програмний вебсервіс для надсилання товарів з/до Amazon. Ця служба більше не буде підтримуватися компанією Amazon. Уся функціональність цієї служби тепер передана Amazon Marketplace;

в) Amazon Historical Pricing – забезпечує доступ до минулих (історичних) даних про продажі;

г) Amazon Mechanical Turk (Mturk) – управляє невеликими одиницями роботи, розподіленими серед багатьох людей;

д) Amazon Product Advertising API – раніше відомий як Amazon Associates Web Service (A2S) і Amazon E-Commerce Service (ECS). Забезпечує доступ до даних про продукти Amazon і функціонування електронної комерції;

е) Amazon Gift Code On Demand (AGCOD) – для корпоративних клієнтів. Дає змогу компаніям миттєво розподіляти подарункові картки Amazon (подарункові коди) будь-якої вартості, інтегруючи технологію подарункових карток Amazon у програму лояльності клієнтів, програму стимулювання працівників і платформу розподілених виплат;

ж) AWS Partner Network (APN) – надає технологічним партнерам та партнерам–консультантам технічну інформацію та підтримку з продажів і маркетингу для збільшення можливостей бізнесу за допомогою AWS. Запущений у квітні 2012 р., APN складається з партнерів з технологій, включаючи незалежних постачальників програмного забезпечення (ISV), постачальників засобів, платформ тощо. Консалтингові партнери включають системних інтеграторів (SI), агентства, консультантів, провайдерів керованих послуг (MSP) тощо. Потенційні технологічні та консультаційні партнери мають відповідати вимогам технічного та нетехнічного навчання, встановленим AWS;

и) Amazon Lumberyard – безкоштовний ігровий двигун AAA-класу, інтегрований з AWS;

к) Amazon Chime – це корпоративне сервісне агентство, яке можна використовувати для голосових повідомлень, відеоконференцій і миттєвих повідомлень.

1.2 Практична частина

1 Необхідно виконати аналіз можливостей віртуалізації відомих вендорів хмарних послуг, зокрема **Google**, **Amazon**, Microsoft [8–14].

2 Необхідно зареєструватися на офіційних сайтах послуг віртуалізації вендорів і провести базові настройки тріал–версій різних сервісів [8–14]:

- спільна робота з документами в Google Disk та Microsoft One Drive;
- розгорнення SaaS–додатків у віртуальному середовищі;
- розгорнення PaaS–додатків у віртуальному середовищі;
- розгорнення IaaS–додатків у віртуальному середовищі.

3 Проаналізувати можливості обраного віртального сервісу вендора хмарних послуг Amazon.

1.3 Зміст звіту

Звіт оформляється кожним здобувачем індивідуально і має містити;

– титульний аркуш з номером і назвою роботи;

– мету роботи;

– результати виконання лабораторної роботи;

– скріншоти за результатами проведеної роботи;

– висновки з роботи.

Контрольні запитання

1 Які собливості побудови архітектури систем віртуалізації Google?

2 Які особливості побудови архітектури систем віртуалізації Microsoft Azure?

3 Які собливості побудови архітектури систем віртуалізації Amazon?

4 Наведіть приклади віртуалізованих рішень і сервісів вендора хмарних послуг Google.

5 Які приклади віртуалізованих рішень і сервісів вендора хмарних послуг Microsoft Azure?

6 Які приклади віртуалізованих рішень і сервісів вендора хмарних послуг Amazon?

7 Які класи в додатку відповідають за обробку запитів?

8 Які методи потрібно реалізувати в класі обробника запитів?

9 Який клас відповідає за список оброблюваних адрес у додатку?

10 Для чого використовують модуль users?

11 У якому класі міститься інформація, передана з HTML-форми?

12 Який клас використовують для зберігання інформації в базі даних об'єктів?

13 Для чого використовується файл шаблону?

Лабораторна робота 2. Ознайомлення з SAAS-, PAAS- ТА IAAS-схемами організації хмарних обчислень

Мета роботи - навчитися виконувати інсталяцію найбільш популярних систем віртуалізації операційних систем (ОС (англ. OS), серверних платформ і систем вкладеної віртуалізації (багаторівневої).

2.1 Методичні вказівки щодо організації самостійної роботи здобувачів

VMware Horizon View – продукт для віртуалізації персональних комп'ютерів, що містить ряд компонентів для організації робочого середовища, даючи змогу централізовано управляти розміщеними на сервері робочими станціями. Horizon View входить до складу бандла Horizon, який у свою чергу ліцензується в трьох варіантах: Standard, Advanced і Enterprise Edition. І хоча ліцензування, як і набір компонентів, які поставляються для кожної з редакцій, є окремою темою, з основних компонентів Horizon View можна виділити такі:

– Connection Server (або Standart Server) – основний компонент і основна роль, яка має бути встановлена в першу чергу. Містить основну консоль управління, де визначаються пули десктопів (desktop pools), додатки, права доступу тощо [1–7];

– Security Server – сервер безпеки, що дає змогу користувачам отримати безпечний доступ до внутрішньої мережі з Інтернету;

– Replica Server – реплікована копія Connection Server. Після того як розгорнутий перший Connection Server, інші вважаються реплікованими. Їхній функціонал полягає в тому, щоб забезпечити резервування, збалансувати навантаження і тим самим підвищити доступність;

– Enrollment Server – використовується для створення True SSO, який дає можливість проходити перевірку автентичності в Microsoft Windows, зберігаючи при цьому всі привілеї домену, при цьому не вимагаючи від

користувачів надавати облікові дані Active Directory. True SSO є технологією VMware Horizon, яка об'єднує VMware Identity Manager 2.6 з Horizon 7. VMware Identity Manager Standard виходить у редакції VMware Horizon 7 Advanced і Enterprise;

– Composer Server – дає змогу використовувати linked-клони і розбиття даних на шари;

– View Agent – дає змогу використовувати робочу станцію або сервер як «джерело»;

– VirtualBox – це програма віртуалізації для операційних систем, розроблена німецькою фірмою Innotek, зараз вона належить Oracle Corporation. Вона встановлюється на наявну операційну систему, яка називається хостовою, усередину цієї програми встановлюється інша операційна система, яку називають гостьовою операційною системою.

Підтримується основними ОС такими, як Linux, FreeBSD, Mac OS X, OS/2 Warp, Microsoft Windows, які підтримують роботу гостьових ОС FreeBSD, Linux, OpenBSD, OS/2 Warp, Windows і Solaris.

Microsoft Hyper-V – система апаратної віртуалізації для x64-систем на основі гіпервізора. Бета-версія Hyper-V була включена в x64-версії Windows Server 2008, а закінчена версія (автоматично, через Windows Update) була випущена 26 червня 2008 р. Раніше була відома як віртуалізація Windows Server (Windows Server Virtualization).

Hyper-V існує у двох варіантах [1–7]:

– як окремий продукт Microsoft Hyper-V Server. Існує чотири версії: Hyper-V Server 2012 R2 (поточна версія Hyper-V), Hyper-V Server 2012, Hyper-V Server 2008 R2 і Hyper-V Server 2008;

– як роль Windows Server 2012 R2, Windows Server 2012, Windows Server 2008 R2, Windows Server 2008 і x64-бітна Pro і Enterprise версії Windows 8, Windows 8.1, Windows 10.

Окрема версія Hyper–V Server є безкоштовною. Перша версія була випущена 1 жовтня 2008 р., є базовим (Server Core) варіантом Windows Server 2008, тобто має повну функціональність Hyper-V; інші ролі

Windows 2008 Server відключені, також лімітовані служби Windows [4]. Безкоштовна 64-бітна Core-версія Hyper-V обмежена інтерфейсом командного рядка (CLI PowerShell), де конфігурація поточної ОС, фізичного апаратного і програмного обладнання виконується за допомогою команд оболонки. Нове меню інтерфейсу управління дає можливість виконати просту первинну конфігурацію, а деякі вільно поширювані скрипти розширюють цю концепцію. Адміністрування та конфігурування віртуального сервера (або гостьових ОС) здійснюється за допомогою програмного забезпечення, встановленого на ПК під управлінням Windows Vista, Windows 7 або Windows 2008 Server зі встановленим додатком для адміністрування Hyper-V з MMC. Іншим варіантом адміністрування / конфігурації сервера Windows 2008 Core є використання віддаленої Windows або Windows Server при перенаправленні (деякої) консолі управління (MMC), що вказує на Core Server. Це значно спрощує налаштування, зводячи його до декількох кліків миші.

У Windows Server 2016 включена оновлена версія Hyper-V. Hyper-V підтримує розмежування згідно з поняттям розділу. Розділ – логічна одиниця розмежування, підтримувана гіпервізором, у якому працюють операційні системи. Кожен екземпляр гіпервізора повинен мати один батьківський розділ із запущеною Windows Server 2008. Стек віртуалізації запускається на батьківському розділі і має прямий доступ до апаратних пристроїв. Потім до основного розділу породжуються дочірні розділи, на яких і розташовуються гостьові ОС. Дочірній розділ також може породити власні дочірні розділи. Батьківський розділ створює дочірні за допомогою API-гіпервізора, наданого в Hyper-V [1–7].

Віртуалізовані розділи не мають ні доступу до фізичного процесора, ні можливості управляти його реальними перериваннями. Замість цього в них є віртуальне подання процесора, і гостьова віртуальна адреса, що залежить від конфігурації гіпервізора, зовсім необов'язково при цьому займає все ВАП. Гіпервізор може визначати підмножину процесорів для

кожного розділу. Гіпервізор управляє перериванями процесора і перенаправляє їх у відповідний розділ, використовуючи логічний контролер штучних переривань (Synthetic Interrupt Controller або скор. SynIC). Hyper-V може апаратно прискорювати трансляцію адрес між різними гостьовими віртуальними адресними просторами за допомогою IOMMU (I/O Memory Management Unit – пристрій управління введенням / виведенням пам'яті), працює незалежно від апаратного управління пам'яттю, використовуваною процесором.

Дочірні розділи не мають безпосереднього доступу до апаратних ресурсів, але натомість отримують віртуальне уявлення ресурсів, називане віртуальними пристроями. Будь-яка спроба звернення до віртуальних пристроїв перенаправляється через VMBus до пристроїв батьківського розділу, які і оброблять цей запит. VMBus – це логічний канал, який здійснює взаємодію між розділами. Відповідь повертається також через VMBus. Якщо пристрої батьківського розділу також є віртуальними пристроями, то запит буде передаватися далі, поки не досягне такого батьківського розділу, де він отримає доступ до фізичних пристроїв. Батьківські розділи запускають провайдер сервісу віртуалізації (Virtualization Service Provider, VSP), який з'єднується з VMBus і обробляє запити доступу до пристроїв від дочірніх розділів. Віртуальні пристрої дочірнього розділу працюють з клієнтом сервісу віртуалізації (Virtualization Service Client або скор. VSC), який перенаправляє запит через VMBus до VSP батьківського розділу. Цей процес прозорий для гостьової ОС.

Віртуальні пристрої також підтримують технологію Windows Server Virtualization, що називається прогресивним введенням / виведенням, для накопичувачів, мережевих і графічних підсистем у тому числі. Enlightened I/O – спеціалізована віртуалізаційна реалізація високорівневих протоколів, як SCSI, для можливості працювати з VMBus безпосередньо, що дає можливість паралельно обробляти будь-які рівні емуляції пристрою. Це робить взаємодію більш ефективною, але натомість вимагає від гостьової

ОС підтримки Enlightened I/O. Тільки Windows Server 2008 R2, Windows Server 2008, Windows 10, Windows 8, Red Hat Enterprise Linux і SUSE Linux зараз мають підтримку Enlightened I/O, що дає їм змогу працювати швидше як гостьовим ОС під Hyper-V порівняно з іншими ОС, яким потрібна більш повільна емуляція пристроїв [1–7].

2.2 Практична частина

2.2.1 Установлення і розгортання VMware Horizon View

VMware Horizon View – продукт для віртуалізації персональних комп'ютерів, що містить ряд компонентів для організації робочого середовища, даючи змогу централізовано управляти розміщеними на сервері робочими станціями. Horizon View входить до складу бандла Horizon, який у свою чергу ліцензується в трьох варіантах: Standard, Advanced і Enterprise Edition. І хоча ліцензування, як і набір, поставляють компоненти для кожної з редакцій і є окремою темою, з основних компонентів Horizon View можна виділити такі:

– Connection Server (або Standart Server) – основний компонент і основна роль, яка має бути встановлена в першу чергу. Містить основну консоль управління, де визначаються пули десктопів (desktop pools), додатки, права доступу тощо [8–14];

– Security Server – сервер безпеки, що дає можливість користувачам отримати безпечний доступ до внутрішньої мережі з Інтернету;

– Replica Server – репліковані копія Connection Server'a. Після того як розгорнутий перший Connection Server, інші вважаються репліками. Їхній функціонал полягає в тому, щоб забезпечити резервування, збалансувати навантаження і тим самим підвищити доступність;

– Enrollment Server – використовують для створення True SSO, який надає можливість проходити перевірку автентичності в Microsoft Windows, зберігаючи при цьому всі привілеї домену, не вимагаючи від користувачів

надавати облікові дані Active Directory. True SSO є технологією VMware Horizon, яка об'єднує VMware Identity Manager 2.6 з Horizon 7. VMware Identity Manager Standard виходить у редакції VMware Horizon 7 Advanced і Enterprise;

– Composer Server – дає змогу використовувати linked-клони і розбиття даних на шари. View Agent дає змогу використовувати робочу станцію або сервер як джерело [8–14].

Підготовка інфраструктури. Для розгортання Horizon View 7 потрібен, як мінімум, один фізичний сервер – хост зі встановленим ESXi версії 6.0 або вище (на момент тестування від створення відмов кластера VMware з двох і більше хостів можна відмовитися). Далі на хості ESXi розгортаються необхідні компоненти Horizon View, більшість з яких можуть так само розміщуватися і на операційних системах, встановлених безпосередньо на залізо (без установлення гіпервізора). Управління середовищем Horizon View 7 можливе тільки в домені Active Directory, і в той же час керуючі компоненти не можуть бути встановлені на контролерах домену [8–14].

Нижче буде розглянуто процес установлення компонентів Horizon View 7.0.1 з використанням гіпервізора VMware ESXi 6.0U2.

У першу чергу необхідно встановити Connection Server і Security Server (якщо доступ з інтернету у внутрішню мережу не потрібний, Security Server можна не встановлювати). Для цього будуть потрібні дві віртуальні машини (ВМ) з Windows Server 2008R2 або Windows Server 2012R2, введені в домен Active Directory. Відповідно буде потрібна, як мінімум, ще одна ВМ, яка виконує функцію контролера домену Active Directory.

Спочатку можна використати три ВМ Windows Server 2012R2 зі статичними IP-адресами, 2vCPU і 4 Гбайт оперативної пам'яті на кожну:

– vconnect.domain.local (192.168.3.230) – Horizon View Connection Server;

– vsecurity.domain.local (192.168.3.231) – Horizon View Security Server;

– dc–01.domain.local (192.168.3.241) – контролер домену Active Directory.

Більш конкретні системні вимоги для установлення кожного з компонентів Horizon View можуть варіюватися від об’єму системи, що масштабується (дивись нижче для кожного компонента).

Для Horizon View так само буде потрібний vCenter Server, який аналогічно не встановлюється на контролері домену Active Directory. Можна використовувати linux-версію – vCenter Server Appliance, установлення якої було розглянуто тут:

– vcsa-01.vdi.local (192.168.3.243) – vCenter Server Appliance.

Зверніть увагу, що доменний контролер Active Directory має стартувати до того, як стартує vCenter Server [8–14].

Установлення Connection Server. Для установлення Horizon View Connenction Server 7.0.1 потрібно [8–14]:

– процесор не нижче Pentium IV 2.0GHz (бажано 4 ядра або 4 vCPU);

– мережевий адаптер: 100 Мбайт/с (бажано 1 Гбайт/с) підключений фізично і доступний через мережу для доменного контролера Active Directory (контролер AD встановлюється на окрему операційну систему);

– оперативна пам'ять 4 Гбайт (для нормальної роботи 50 і більш віддалених робочих столів рекомендується 10 Гбайт і більше);

– операційна система Windows Server 2008 R2 SP1 (Standart, Enterprise або Datacenter) або Windows Server 2012 R2 (Standard або Datacenter). Починаючи з Horizon View версії 7.0.3. підтримується Windows Server 2016.

Перш ніж приступити до установлення Connection Server, у домені Active Directory domain.com необхідно створити групу адміністраторів Horizon View, яка включає, як мінімум, одного користувача, наприклад viewadmin. Для запобігання проблемам з нестачею прав, треба зробити адміністраторів Horizon View (Horizon View Admins (рисунок 2.1))

членами групи адміністраторів домену. Отже, група адміністраторів Horizon View є адміністраторами домену domain.local. Про всяк випадок треба проводити встановлення всіх компонентів Horizon View не через Remode Desktop, а безпосередньо через консоль гіпервізора (у vSphere Client на ВМ викликати меню по правій кнопці миші і вибрати Open Console) [8–14].

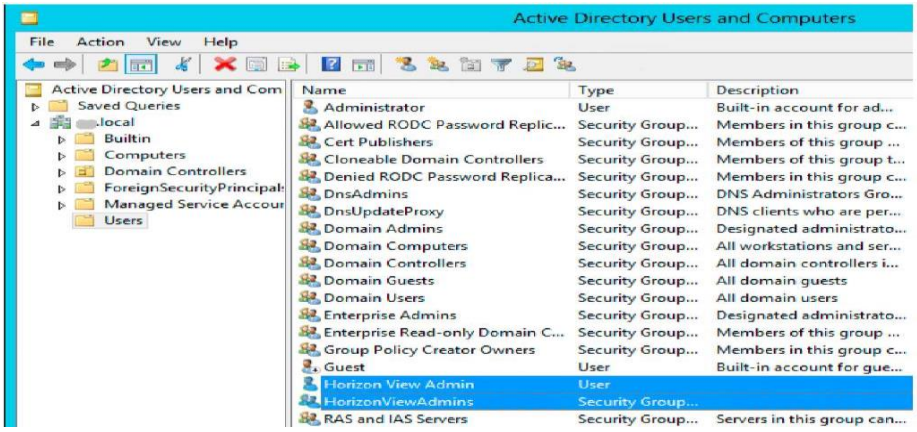


Рисунок 2.1 – Створення групи Horizon View Admins і користувача vewadmin

Логіном під користувачем viewadmin на vconnect.domain.local і запускаємо інсталятор (рисунок 2.2). В інтерактивному режимі вибираємо IPv4, вказуємо пароль для відновлення бекапа (Data Recovery Password) (рисунок 2.3), автоматичну конфігурацію брандмауера Windows (рисунок 2.4) і доменну групу адміністраторів Horizon View: domain.local \ HorizonViewAdmins відкриваємо.

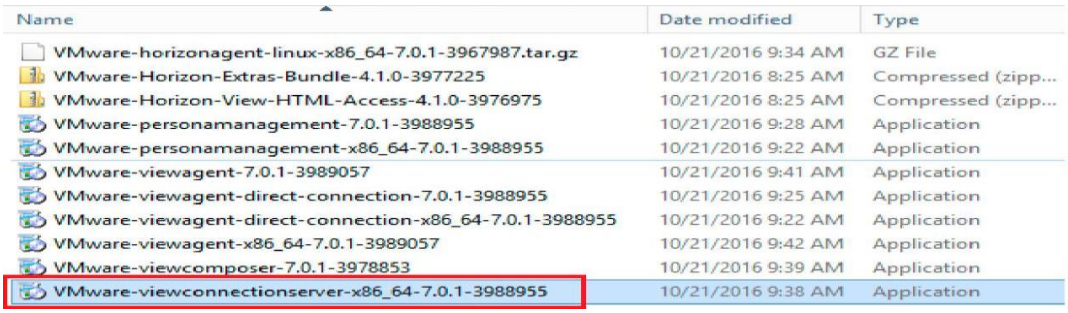


Рисунок 2.2 – Запуск інсталятора Horizon View Connection Server

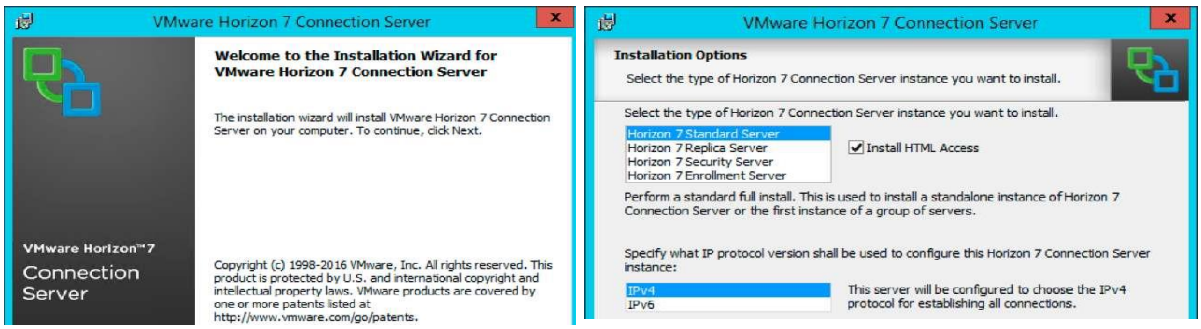


Рисунок 2.3 – Початок установлення Connection Server

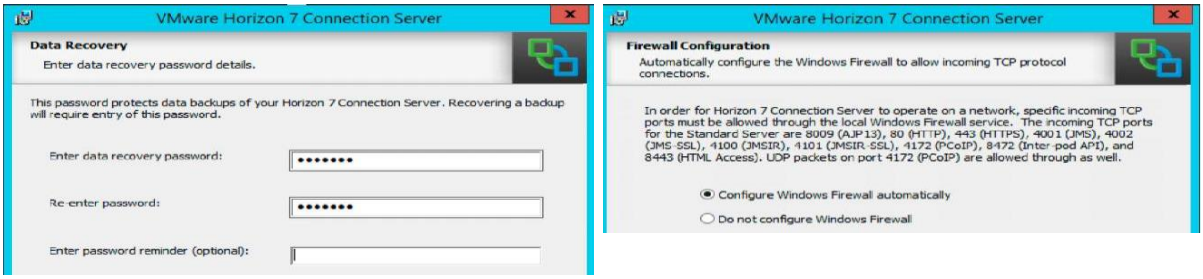


Рисунок 2.4 – Продовження установлення Connection Server

Установлення Security Server. Перед установленням Security Server (рисунок 2.5) необхідно задати pairing password для підключення до Connection Servery, натискаємо на робочому столі Connection Server на ярлик Horizon 7 Administrator Console або в адресному рядку браузера вводимо

https:ll<FQDN_или_IP_Connection_Server'a**>ladmin**

Inventory > View Configuration > Servers > Connection Servers > More Commands > Specify Security Pairing Password...

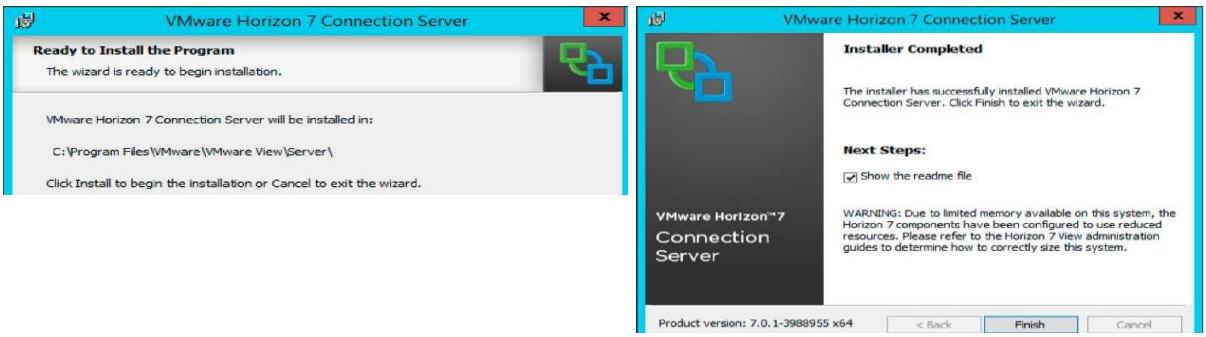


Рисунок 2.5 – Задавання pairing password для підключення до Connection Servery

У вікні двічі вводимо пароль і встановлюємо термін його дії (наприклад 30 хв) (рисунки 2.6 і 2.7). Тепер можна перейти до установлення Security Server, системні вимоги для якого будуть ідентичні Connection Server. Відкриваємо той самий установник, що і для Connection Server (рисунок 2.8), вибираємо Security Server, вказуємо hostname або IP-адресу вже встановленого Connection Server, вводимо пароль і переходимо до налаштування зовнішніх URL (рисунки 2.9 і 2.10).

Рисунок 2.6 – Вхід у вебінтерфейс Horizon 7

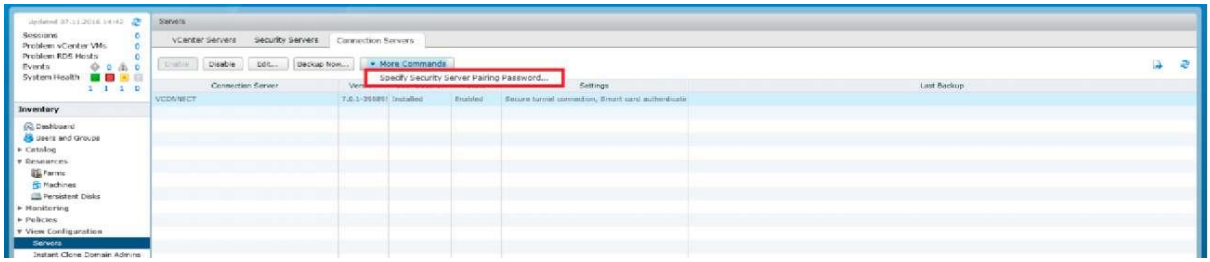


Рисунок 2.7 – Створення Securuty Pairing Password

Рисунок 2.8 – Запуск інстолятора Security Server

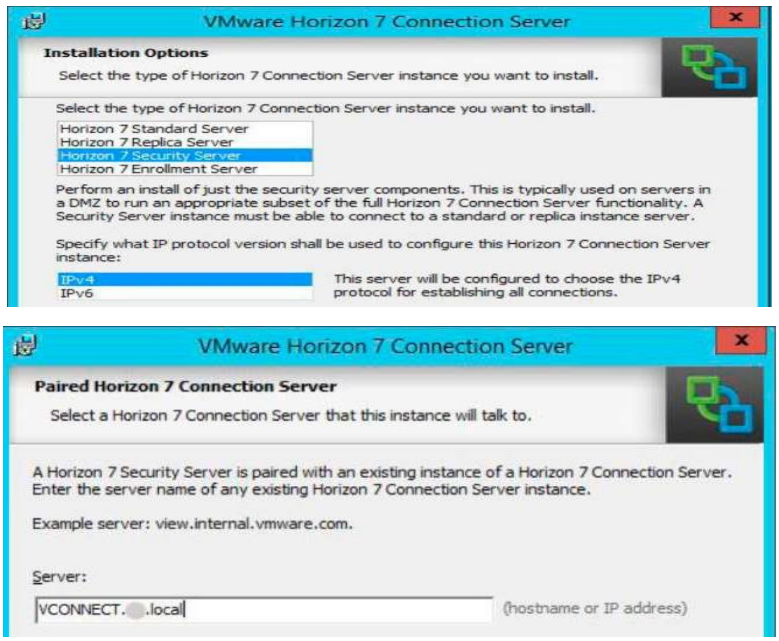


Рисунок 2.9 – Установлення Securuty Server

Як правило, значення підставляються автоматично (рисунок 2.10):

External URL: https://vsecurity.domain.local:443;

PCoIP External URL: 192.168.3.231:4772;

Blast External URL: https://vsecurity.domain.local:8443.

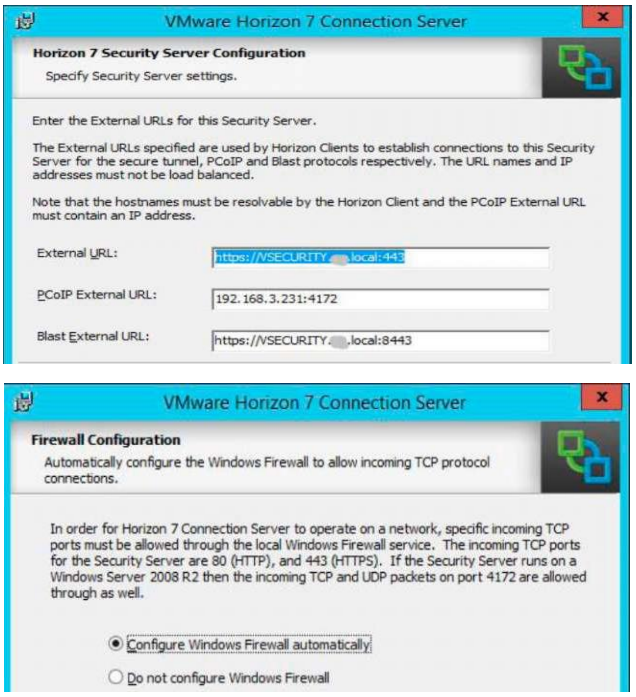


Рисунок 2.10 – Установлення Security Server

Після завершення установлення відкриваємо вебконсоль і переходимо *Inventory* > *View Configuration* > *Servers* > *Security Servers*, де бачимо, що з'явився Security Server (рисунок 2.11).

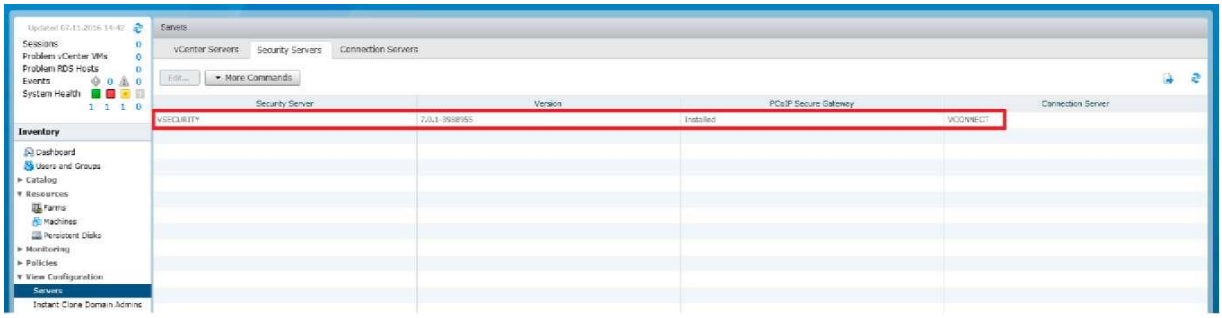


Рисунок 2.11 – Вебконсоль Horizon View: вкладка Servers

Додавання vCenter Server

Наступним етапом необхідно додати vCenter Server:

– відкриваємо вебконсоль і переходимо Servers –> vCenter Servers –> Add і у вікні вказуємо IP-адресу, ім'я користувача і пароль для доступу до vCenter Server (рисунок 2.12);

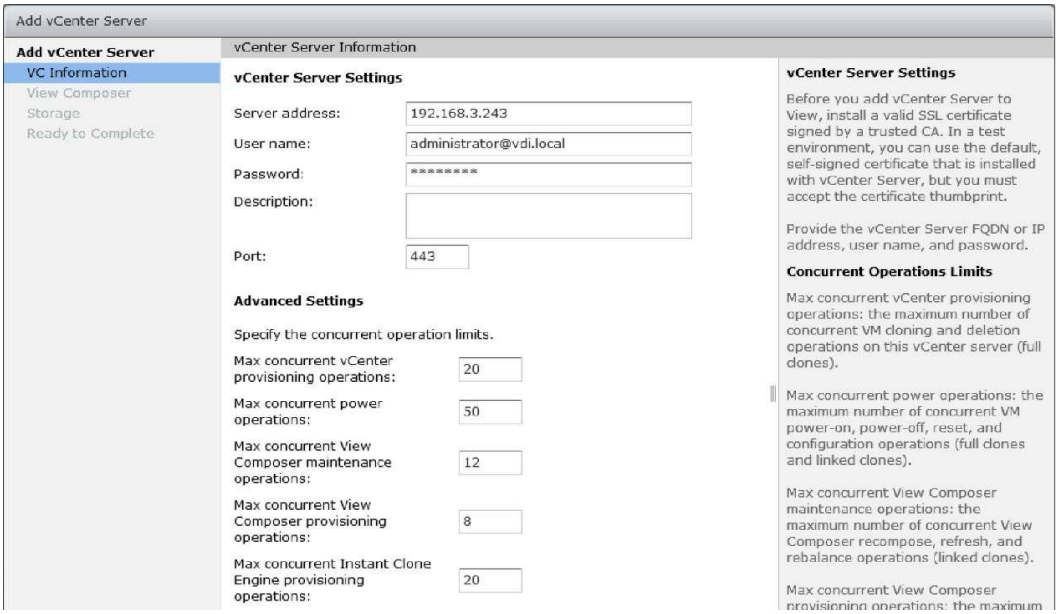


Рисунок 2.12 – Додаток vCenter Server

– Max concured vCenter provosioning operations – визначає максимальну кількість створюваних і видалених клонів ВМ (про створення повних клонів ВМ піде далі) для підключення vCenter Server;

– Max concured power on operations – визначає максимальну кількість операцій power-on, power-of– і reset для повних клонів (Full clones) і пов'язаних клонів (Linked Clones), робота з якими буде розглянута далі;

– View Composer Maintance Operations – максимальна кількість операцій recompose, refresh і rebalance для Composer Server, який підключати поки що не будемо;

– Max concurent Instant Clone Engine Operations – визначає максимальну кількість операцій для підключення vCenter Server для створення і видалення пов'язаних клонів (Linked Clones).

Поки що всі параметри можна залишити в їхніх дефолтних значеннях. Ігноруємо повідомлення про некоректні сертифікати (повідомлення, скоріше за все, з'явилося, тому що налаштування сертифікатів для тестових середовищ необов'язкове і буде розглянуто пізніше) і переходимо до налаштувань View Composer Server, який поки що не встановлено. Вибираємо Do not use View Composer (рисунок 2.13).

Хости ESXi, керовані vCenter Server'oM, можна конфігурувати для кешування дисків віртуальних машин. Ідея полягає в тому, щоб зменшити кількість операцій введення / виведення за рахунок кешування хостами найбільш використовуваних ділянок даних. Починаючи з vSphere 5.x віртуальні машини можуть бути налаштовані для звільнення невикористовуваних ділянок дисків (наприклад область віддалених файлів). На етапі тестування або при малих навантаженнях залишаємо View Storage Accelerator ввімкненим, а інші налаштування залишаємо за замовчуванням (рисунки 2.14 і 2.15).

Після завершення відкриваємо вебконсоль Horizon і переходимо в *View Configuration –> Servers –> вкладка vCenter Servers*, де видно, що з'явилася інформація про підключений vCenter Server (рисунок 2.16).

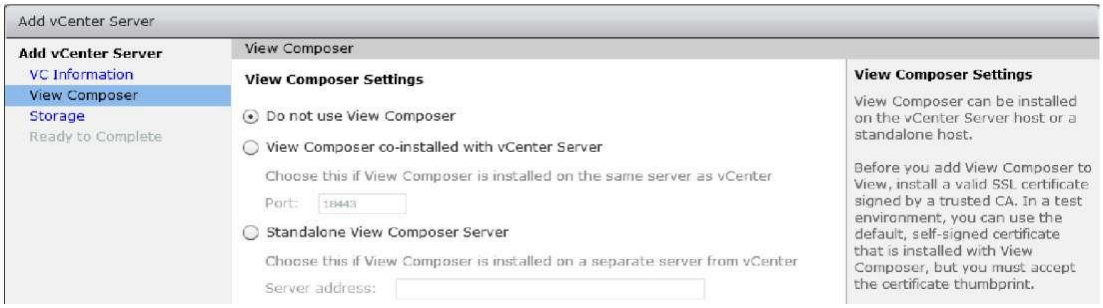


Рисунок 2.13 – Додаток vCenter Server: вкладка View Composer

Рисунок 2.14 – Додаток vCenter Server: вкладка Storage

Рисунок 2.15 – Додаток vCenter Server

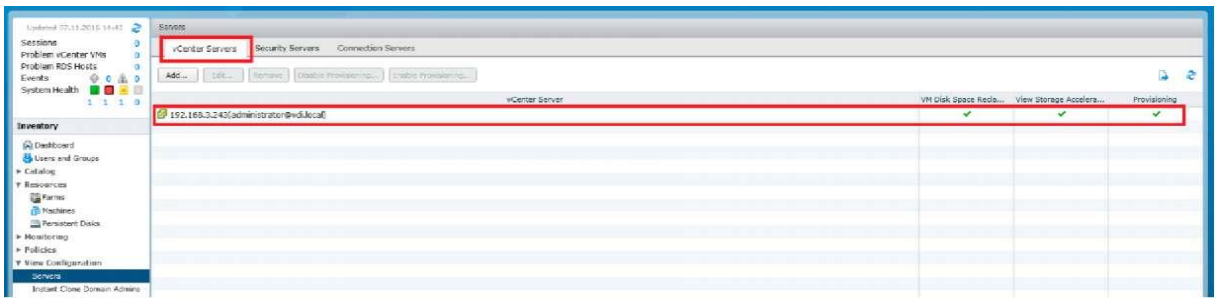


Рисунок 2.16 – Вебконсоль Horizon View: вкладка Servers

Активація ліцензії Horizon View

Для активації ліцензії Horizon View необхідно відкрити вебконсоль Horizon, перейти в розділ активації ліцензії (рисунок 2.17).

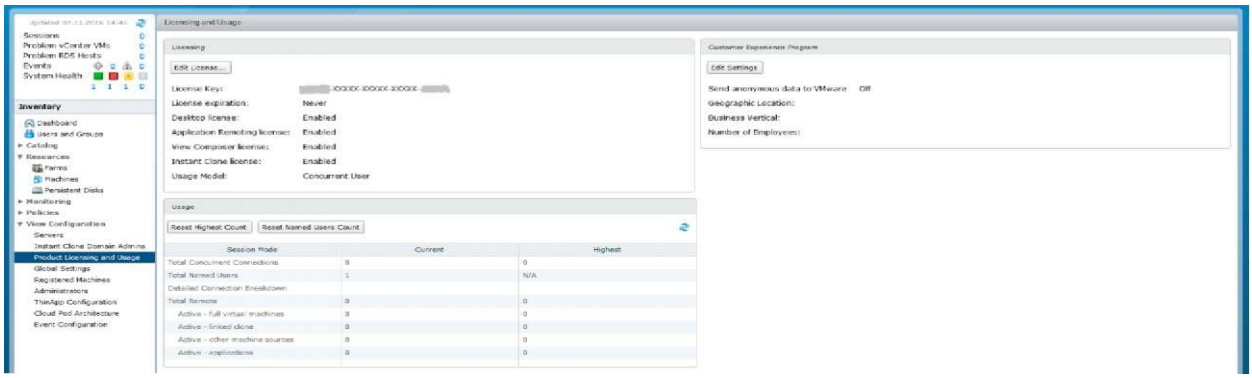


Рисунок 2.17 – Активація ліцензії Horizon

2.2.2 Створення та підключення Events Database

Horizon View може використовувати підключену систему управління базами даних (СУБД) для журналювання подій, наприклад призначених для користувача підключення. Можна використовувати як Oracle, так і Microsoft SQL Server.

Нижче буде розглянуто приклад використання Microsoft SQL Server 2016 Developers Edition, але можна скористатися і безкоштовною редакцією Microsoft SQL Server 2012 Express з максимальним розміром реляційної бази даних **(БД)** до 10 Гбайт і підтримкою до одного процесора або чотирьох ядер. Якщо ви встановлюєте редакцію Express, то при додаванні інстанції бази даних у вкладці Instance Configuration слід примусово вибрати Default instance (або MSSQLSERVER). В іншому випадку Horizon View не зможе підключитися до створеної інстанції MS SQL.

Перед установленням системи управління базами даних **(СУБД)** бажано так само створити знімок (shanpshot) віртуальної машини, щоб в разі помилок у конфігурації (наприклад недефолтне Instance name) можна було повернутися в стан системи до установлення СУБД, не виконуючи

довгу процедуру видалення MS SQL: View Configuration –> Product Licensing and Usage –> Edit License і ввести серійний ключ (рисунок 2.17).

Установлення СУБД MS SQL буде проводитися на віртуальній машині, де розташований Security Server (vsecurity.domain.local), але з метою економії ліцензій ОС Windows усі компоненти СУБД можна так само встановлювати разом з Connection Server. Віртуальної машини з 4 Гбайт оперативної пам'яті, двома ядрами процесора і 15 Гбайт додаткового місця на жорсткому диску буде цілком достатньо. За більш детальною інформацією звертайтеся до системних вимог конкретних версій Microsoft SQL Server.

Після установлення всіх необхідних компонентів Microsoft SQL Server необхідно запустити Microsoft SQL Management Studio та вибрати (рисунок 2.18): БД –> Створити БД ... У вікні вказуємо ім'я створюваної БД (ViewEvents (рисунок 2.19)) і у вкладці параметри задаємо модель відновлення (Recovery Model): проста (рисунок 2.20).

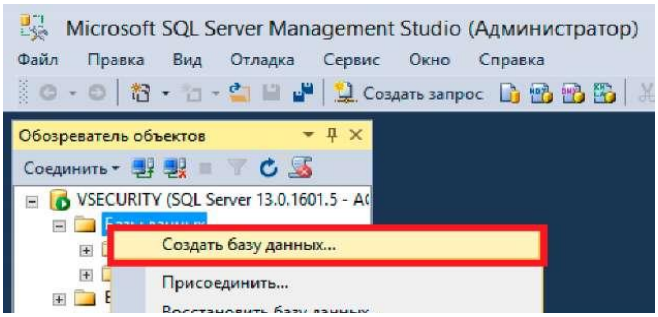


Рисунок 2.18 – Створення нової БД у Microsoft Management Sutio

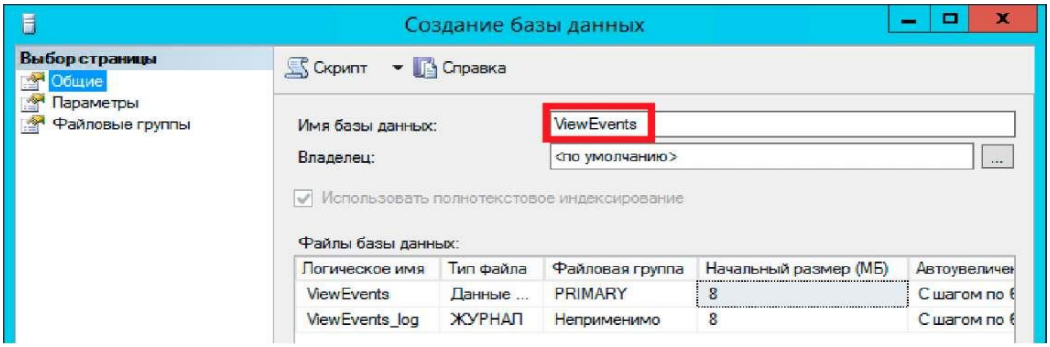


Рисунок 2.19 – Загальні параметри створюваної БД

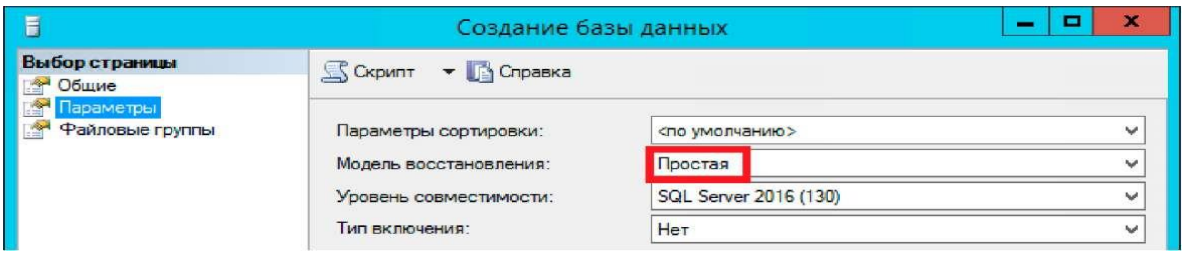


Рисунок 2.20 – Створення нової простої БД

Тепер необхідно створити новий обліковий запис, щоб отримати доступ до цієї БД. Для цього відкриваємо: *Безпека –> Імена входу –> Створити ім'я для входу* ... або для англомовної версії MS SQL: *Security –> Logins –> New Login* і створюємо новий обліковий запис ViewEvents з перевіркою достовірності SQL Server (SQL Server authentication) і обов'язковим відключенням пункту Вимоги, застосовуючи політику паролів (Enforce Password policy). Так само призначаємо користувачеві БД за замовчуванням – ViewEvents (рисунок 2.21).

Переходимо у вкладку Ролі сервера (Server roles) і призначаємо обліковий запис права доступу sysadmin (рисунок 2.22).

У вкладці зіставлення користувачів (User Mapping) зіставляємо БД ViewEvents і призначаємо роль db_owner (рисунок 2.23).

Необхідно так само переконатися, чи створені БД вимикають правила брандмауера (порт 1433), а також треба увімкнути протокол TCP / IP для використовуваної інстанції MS SQL Server. Для цього відкриваємо SQL Server Configuration Manager і переходимо, як зазначено на рисунку 2.24.

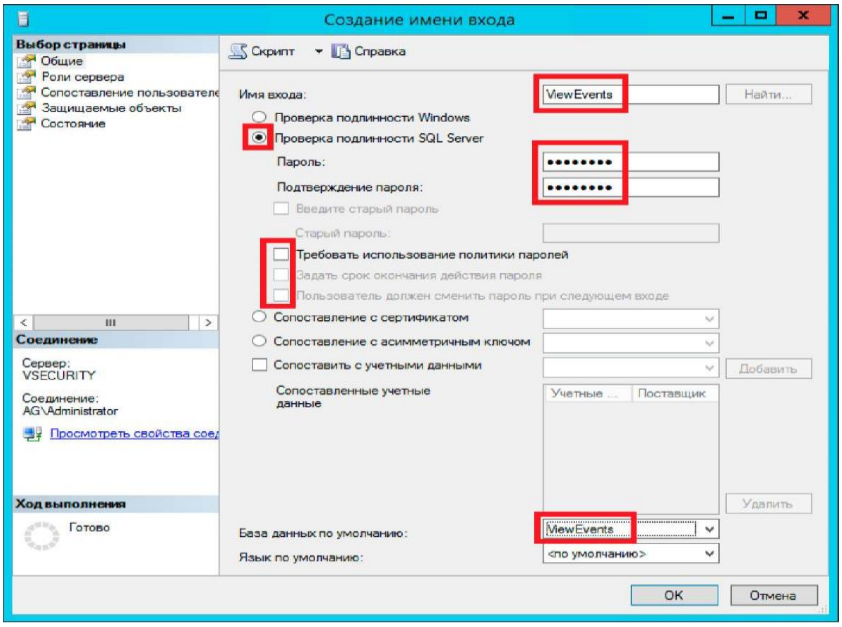


Рисунок 2.21 – Створення нового облікового запису в MS SQL Express

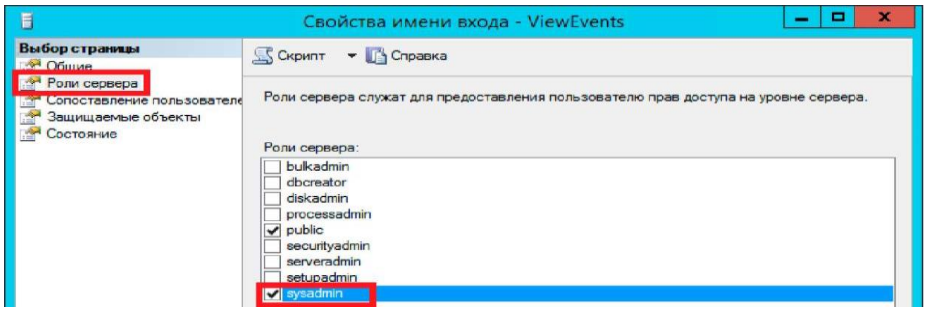


Рисунок 2.22 – Створення нового облікового запису в MS SQL Express

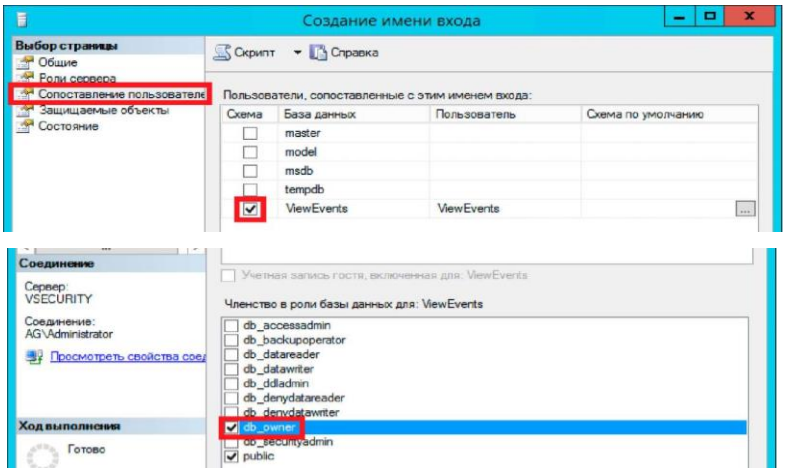


Рисунок 2.23 – Створення нового облікового запису
в MS SQL Express MS SQL Express

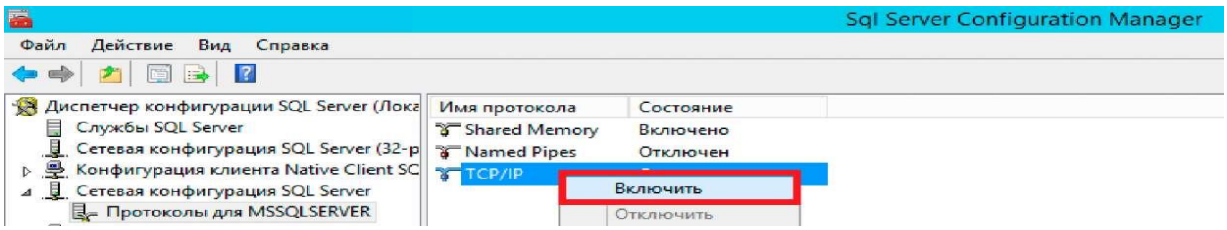


Рисунок 2.24 – SQL Server Configuration Manager:
увімкнення протоколу TCP/IP

Мережева конфігурація SQL Server Протоколи для *{ім'я_інстанції_MS_SQL}> TCP /IP> увімкнути*.

Щоб змінені налаштування вступили в дію, необхідно перезапустити службу SQL Server (ім'я_інстанції_MS_SQL). Повертаємося у вебконсоль Horizon, відкриваємо: *View Configuration –> Event Configuration –> Event Database –> Edit* і у вікні вводимо параметри підключення до БД (рисунок 2.25). Event Settings можна залишити за замовчуванням (рисунок 2.26).

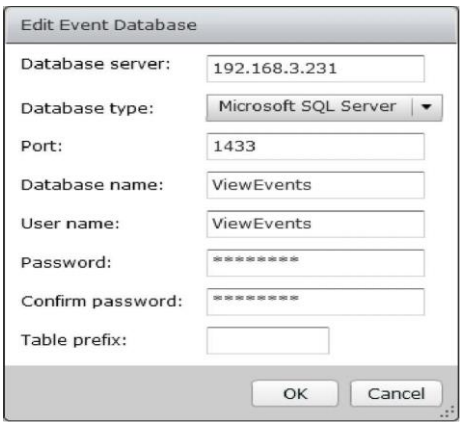


Рисунок 2.25 – Підключення Events DB до Horizon View

Тепер статус Events Database відображується у вебконсолі на Dashboard (рисунок 2.27), а журнал доступний по переходу: *Monitoring –> Events*, як показано на рисунку 2.28.

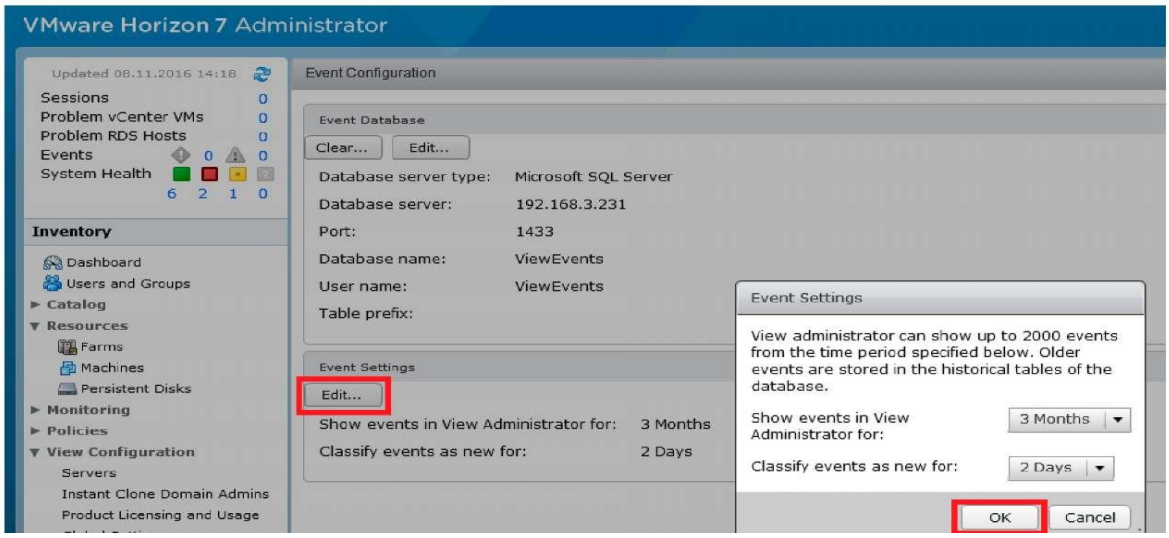


Рисунок 2.26 – Horizon Event Settings

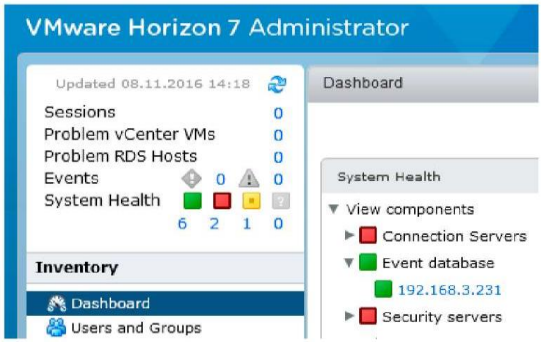


Рисунок 2.27 – Horizon Dashboard

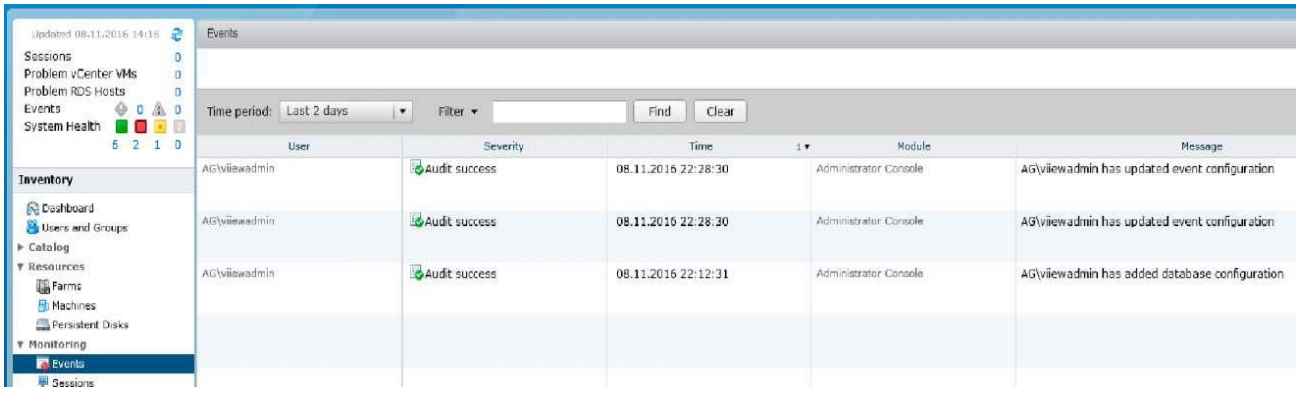


Рисунок 2.28 – Horizon Events

Додавання ферми Microsoft RDS. Підключення ферми Microsoft RDS до Horizon View дає змогу опублікувати як додатки, так і робочі столи

в єдиному середовищі Horizon і організовує доступ до ресурсів RDS через протокол PColP або VMware Blast. Підключення ферми Microsoft RDS є для Horizon у редакціях Advanced і Enterprise [8–14].

Вимоги до створюваної ВМ. ОС Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2 або Windows Server 2016 встановлена на фізичний сервер, або віртуальну машину. Для використання протоколів PColP або VMware Blast необхідно також встановити Horizon Agent; процесор – мінімум 8 vCPU для Windows Server 2012 R2 і 4 vCPU для 2008R2; оперативна пам'ять – для хоста сесій RDS на 8 vCPU рекомендується в середньому 24–48 Гбайт; жорсткий диск – використання Thin provision дасть змогу заощадити місце в datastore, оскільки профілі користувачів (C: \ Users) будуть поступово збільшувати обсяг інформації, що зберігається; знімні накопичувачі – Floppy drive можна одразу ж видалити, а ось що стосується CD / DVD–привода, то його слід демонтувати після установлення ОС і VMware Tools і перемкнути в положення Client Device, який не підключений; мережа – для зменшення network latency як ethernet-адаптера рекомендується вибрати VMXNET 3 (для цього необхідно також встановити VMware Tools з включеною опцією підтримки VMXNET 3) [8–14].

Для більш детальної інформації можна ознайомитися з Horizon RDSH Performance & Best Practices. Створимо ферму RDS rds–01.domain.local і додамо її для доступу через Horizon View. Сам процес розгортання ферми RDS на базі Windows 2012R2 тривіальний і здійснюється через диспетчер серверів (Server Manager) (рисунки 2.29, 2.30).

Далі встановлюємо необхідний для роботи софт (публіковувати його засобами Microsoft RDS не потрібно, оскільки у Horizon View своя консоль управління додатками) і переходимо до установлення View Agents на Session Host (рисунок 2.31): читаємо ліцензійну угоду (рисунки 2.32), підтверджуємо, вибираємо протокол IPv4 і переходимо в меню вибору встановлюваних компонентів (рисунок 2.33).

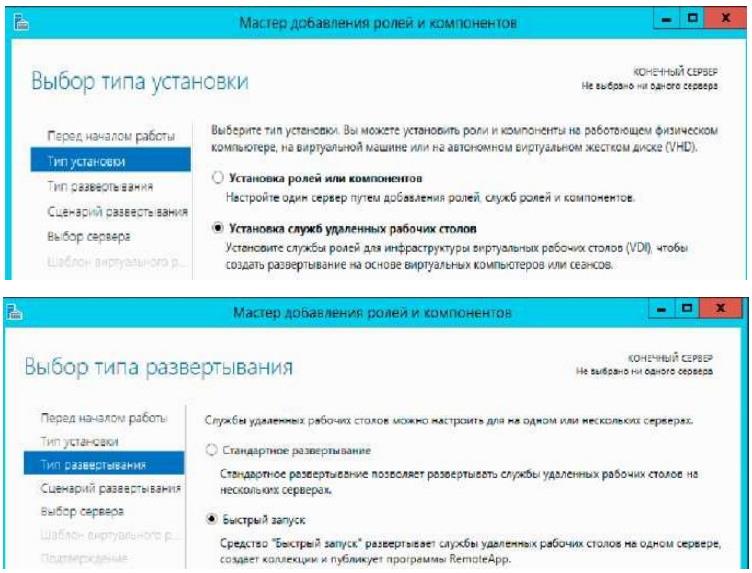


Рисунок 2.29 – Вибір типу установлення та розгортання ферми Microsoft RDS

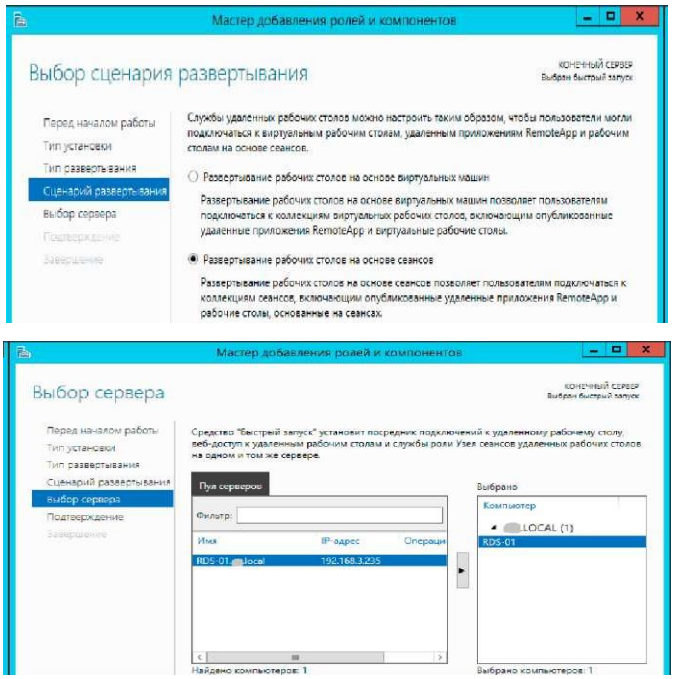


Рисунок 2.30 – Вибір сценарію розгортання ферми та сервісу Microsoft RDS

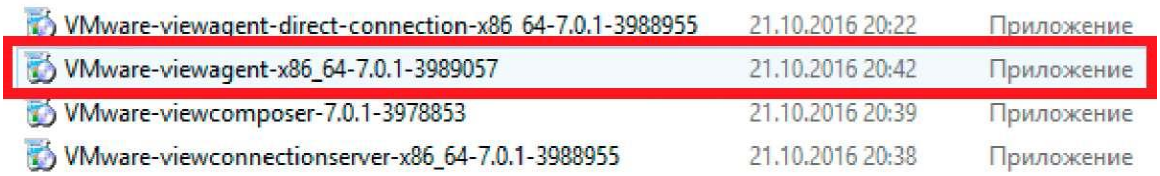


Рисунок 2.31 – Установлення View Agents на Session Host

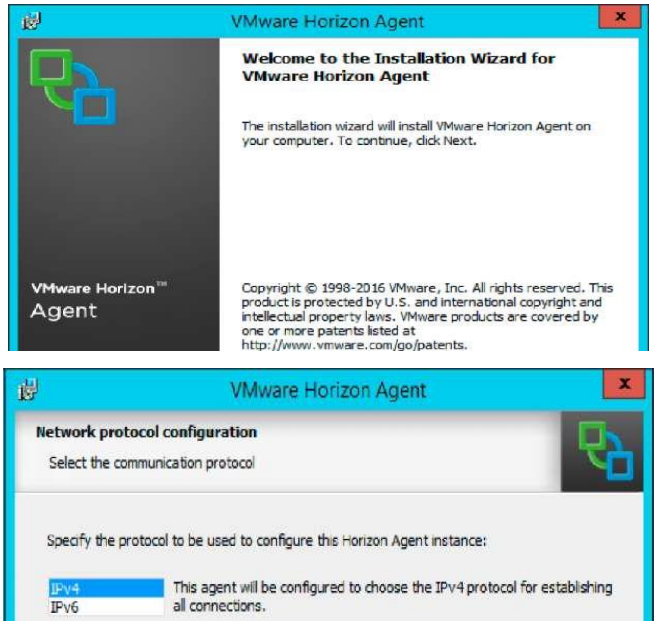


Рисунок 2.32 – Початок установлення View Agent

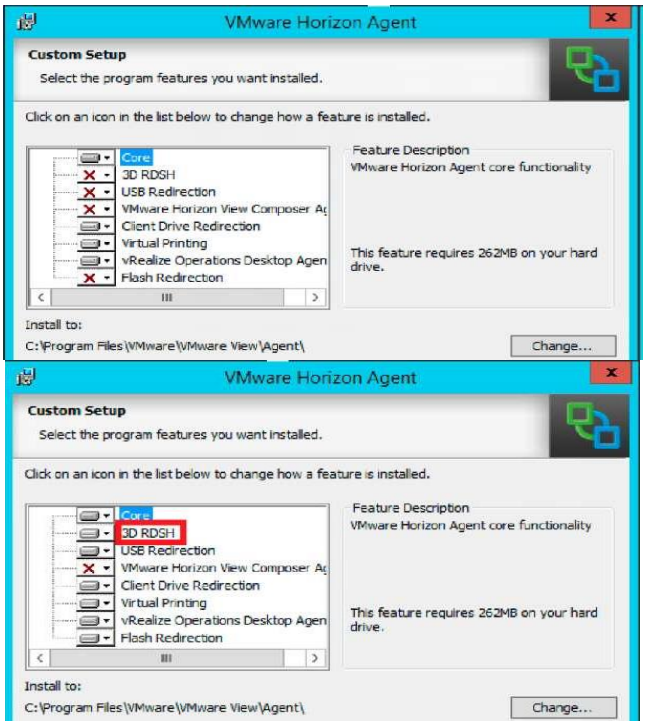


Рисунок 2.33 – Вибір директорії для установлення View Agent

На етапі тестування можна вибрати майже всі компоненти, за винятком VMware Horizon View Composer Agent – для ферм Microsoft RDS він не потрібен. Якщо робота в додатках потребує використання графічної акселерації, то вибираємо 3D RDSH (рисунок 2.33).

Вказуємо ім'я хоста connection Server (vconnect.domain.local), вводимо логін і пароль адміністратора Horizon View (DOMAIN \ viewadmin) і завершуємо налаштування агента після натискання кнопки Install (рисунок 2.34). Вибираємо неавтоматизовану ферму (Manual Farm), задаємо її ID і переходимо до налаштувань підключення ферми (рисунок 2.35): Resources –> Farms –> Add...

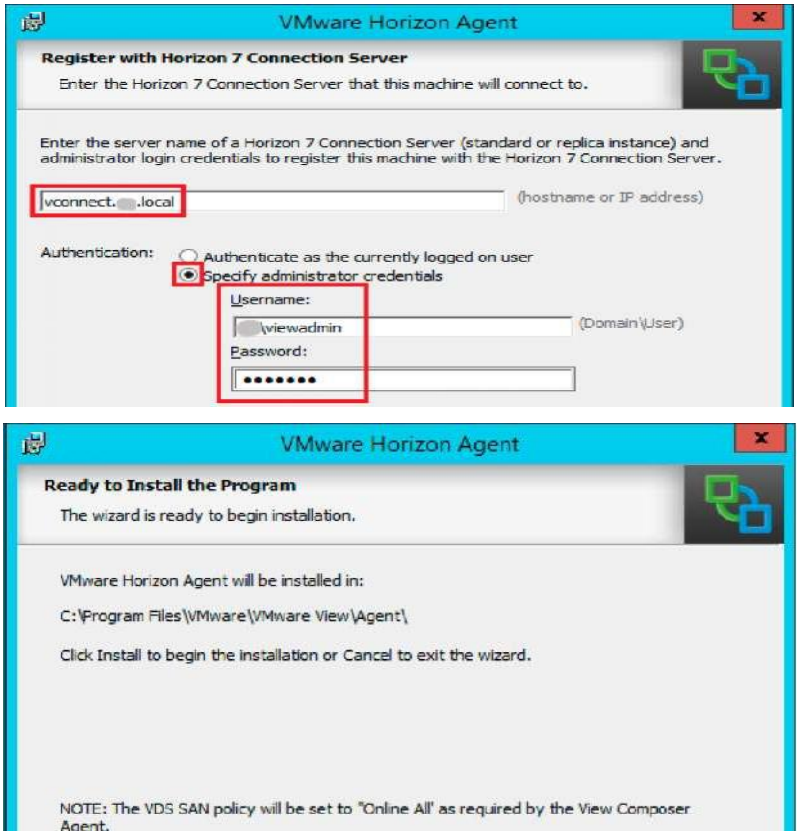


Рисунок 2.34 – Установлення View Agent

Default Display Protocol буде використовуватися під протокол Microsoft RDP, PColP, або з'явився в сьомій версії Horizon View – VMware Blast.

Коротко про відмінності між PColP і VMware Blast: PColP дає більше навантаження на клієнтський пристрій, але використовує меншу ширину каналу, ніж VMware Blast. А VMware Blast використовує велику ширину каналу, але забезпечує більш комфортну роботу користувача (за рахунок підтримки ClearType шрифтів 32-бітовим зображенням тощо) [8–14].

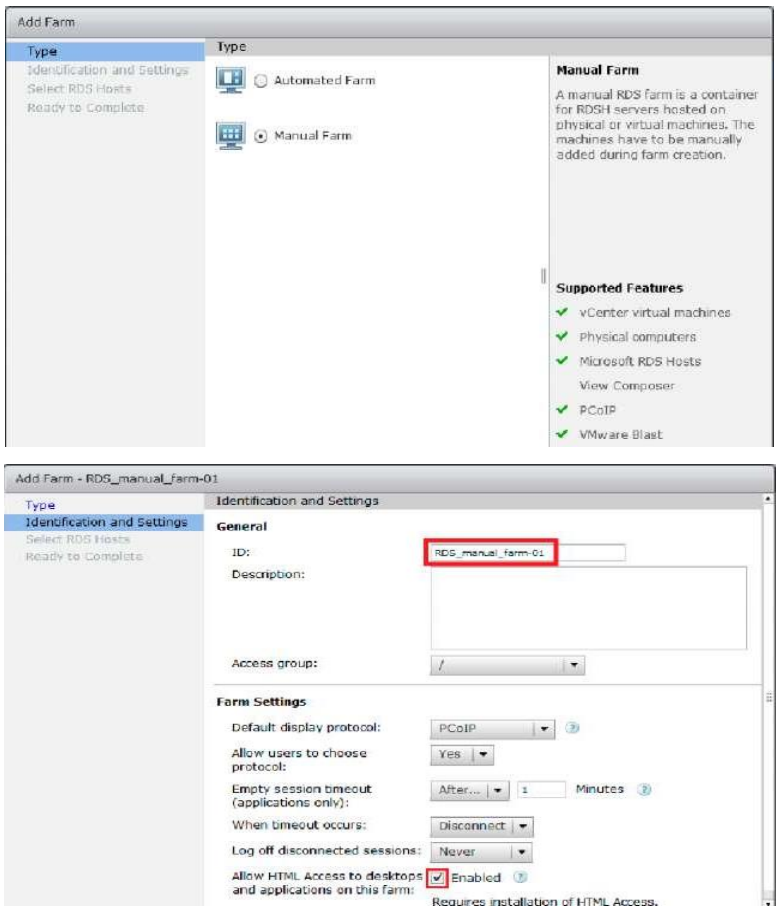


Рисунок 2.35 – Додаток RDS Farm

Allow users to choose protocol – параметр дає змогу користувачам на боці клієнта вибирати протокол. Інші параметри впливають на підключення і призначену для користувача сесію:

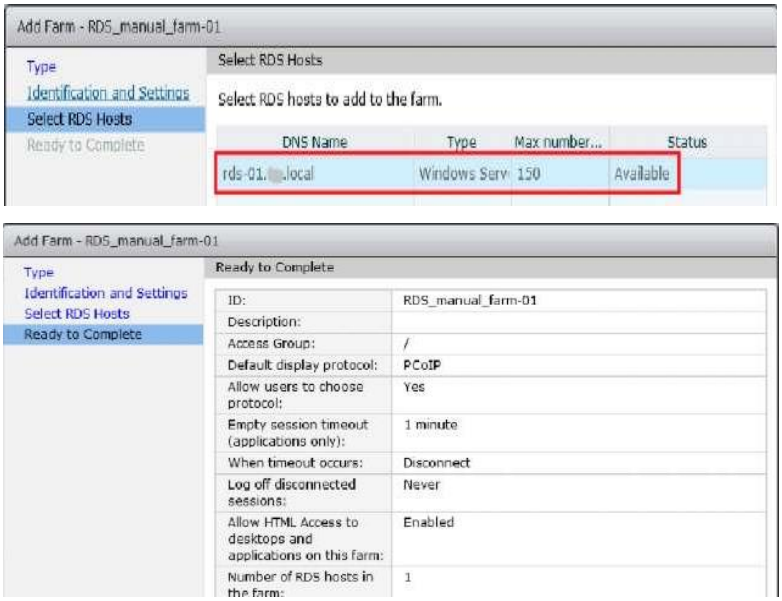
– Empty session timeout – визначає кількість часу, протягом якого порожня сесія додатків (без відкритих додатків) вважається відкритою. Відкрита сесія дає змогу користувачам швидше запустити додаток, у той час як автоматичне їхнє закриття дає можливість заощадити споживані системні ресурси;

– When timeout occurs – означає від'єднання порожнього сеансу додатків або ж вихід після досягнення тайм-ауту сесії додатків. Значення Log off звільняє ресурси, але повторний запуск додатків триватиме довше.

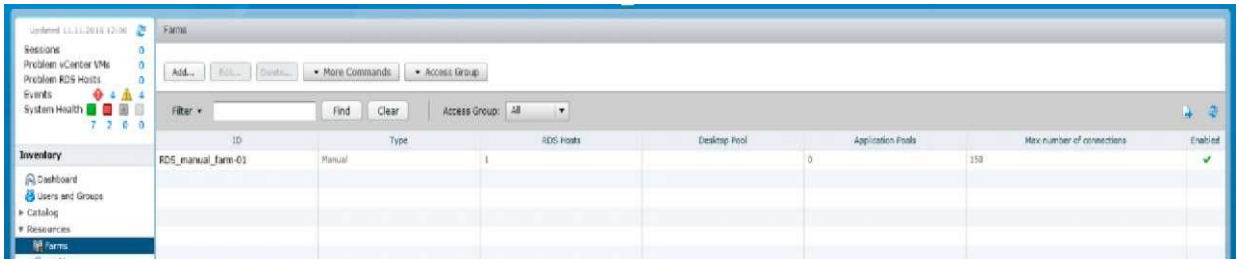
Після установлення відкриваємо в браузері консоль адміністрування Horizon і переходимо: Log off disconnected sessions – відповідно визначає, через скільки здійснювати вихід з системи (Log Off) при роз'єднанні сесії.

Будьте уважні при налаштуванні параметра, щоб уникнути втрати даних, особливо при виборі значення Негайно (Immediate) – роз'єднана сесія буде негайно завершена (Log off) [8–14].

І останній пункт, який дає доступ до десктопів і додатків у створюваній фермі – Allow HTML Access to desktops and applications on this farm (можна використовувати для зручності). Далі підключається RDS-хост і фіналізуємо створення ферми RDS (рисунок 2.36, а). Після закінчення у вебконсолі Horizon у вкладці Farms з'явиться додана ферма Microsoft RDS (рисунок 2.36, б).



а)



б)

Рисунок 2.36 – Додавання RDS Farm і результат підключення Manual Farm

Додавання пулу десктопів Microsoft RDS. Для того щоб опублікувати в Horizon View підключену ферму Microsoft RDS, необхідно відкрити у вебконсолі Horizon [8–14]: Catalogue –> Desktop Pools –> Add.

У майстрі вибираємо RDS Desktop Pool, вводимо ID і коротке ім'я (Display Name) і переходимо до налаштування пулу (рисунок 2.37, а).

Обмеження для Connection Server (Connection Server Restrictions) вибираємо порожніми (None), у такий спосіб підключатися до створюваного пулу можна буде через будь-який Connection Server (поки що він один). Налаштування Adobe Flash залишаємо за замовчуванням (детальніше буде розглянуто трохи нижче).

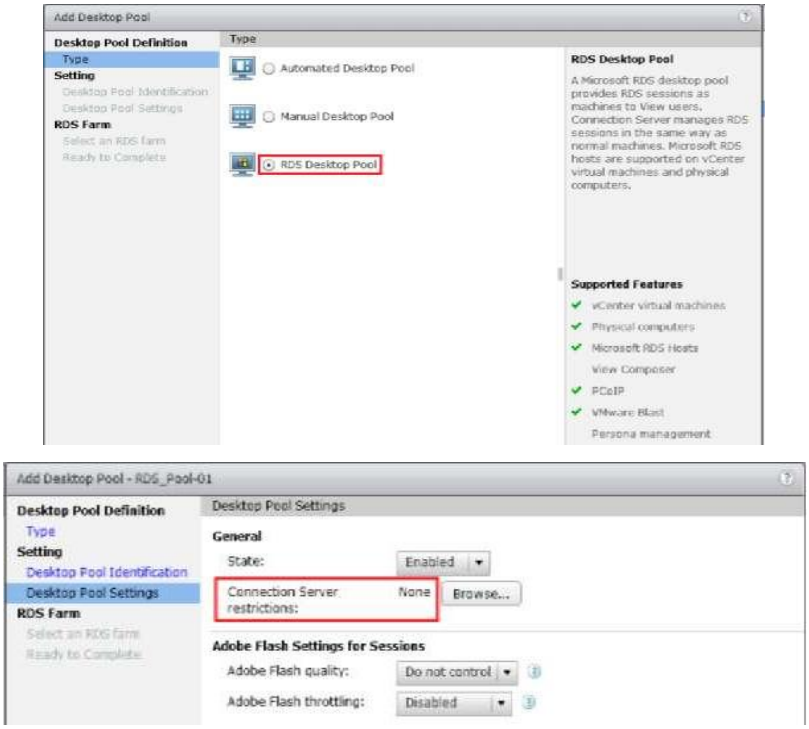
Вибираємо ферму RDS зі списку (рисунок 2.37, б) і на фінальному вікні майстра залишаємо чек-бокс на опції Entitle users after the wizard finishes, після чого у вікні, що відкрилося, тиснемо кнопку Add... і додаємо користувачів, які мають отримати доступ до створюваного RDS Desktop Pool.

Створення Application Pool. Створення пулу додатків дає змогу опублікувати через Horizon View додатки, встановлені на фермі Microsoft RDS. На кожну ферму RDS можна створити тільки один пул додатків. Додатки, опубліковані в пулі, мають бути встановлені ідентично на всіх серверах ферми RDS. Створення пулу дає змогу працювати в редакціях Advanced і Enterprise.

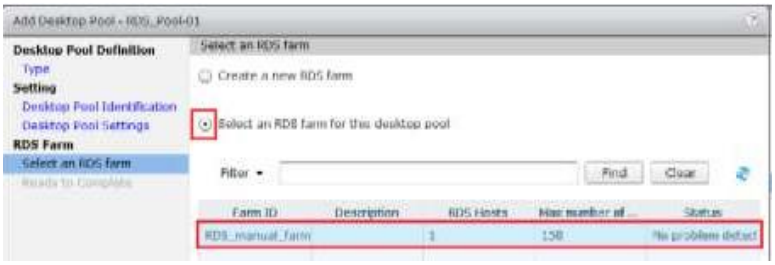
Зверніть увагу, що в Horizon View є свої інструменти для публікації додатків, тому публікувати встановлені на фермі RDS додатки штатними засобами Microsoft RDS необхідності нема [8–14].

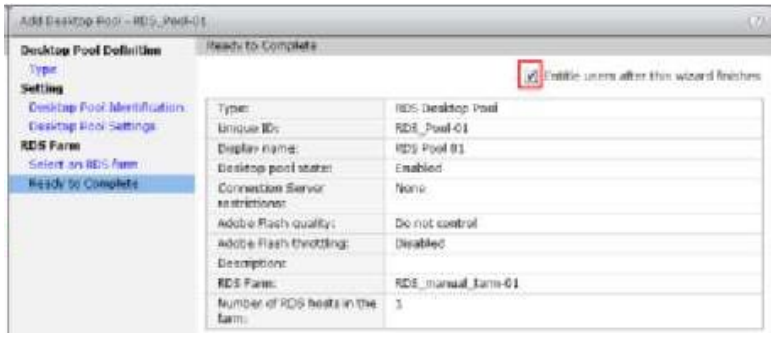
Для створення пулу додатків переходимо: *Catalog –> Application Pools –> Add...*

Вибираємо ферму RDS, вибираємо додатки для публікації, відзначаємо чек-бокс на пункті Entitle users after this wizard finishes (рисунок 2.38) і після закінчення створення пулу переходимо в Add Entitlement, що дає необхідним користувачам Active Directory доступ до пулу додатків (DOMAIN \ User1, User2, User3 ...).



а)





б)

Рисунок 2.37 – Додавання Microsoft RDS Desktop Pool

Використання Horizon View Client для запуску додатків. Тепер усе готово для того, щоб підключитися і запустити опубліковані додатки або підключитися до створеного пулу десктопів Microsoft RDS.

Підключимося до Connection Server (рисунок 2.39), використовуючи Connection View Client. Для цього відкриваємо в браузері: https: // {FQDN або IP Connection Server} [8–14].

Натискаємо на посилання і переходимо до повного списку клієнтів Horizon View для різних операційних систем.

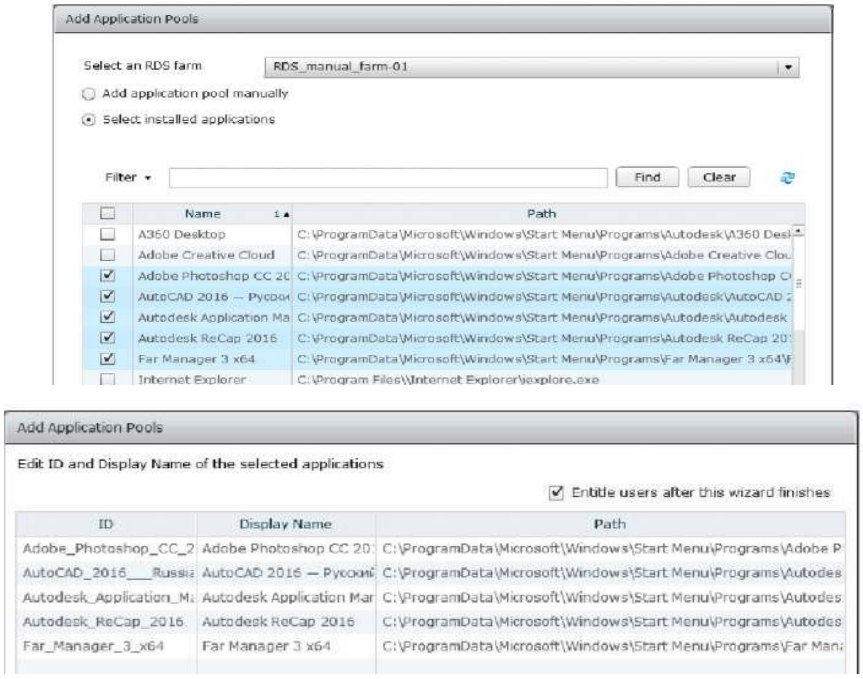


Рисунок 2.38 – Створення Application Pool

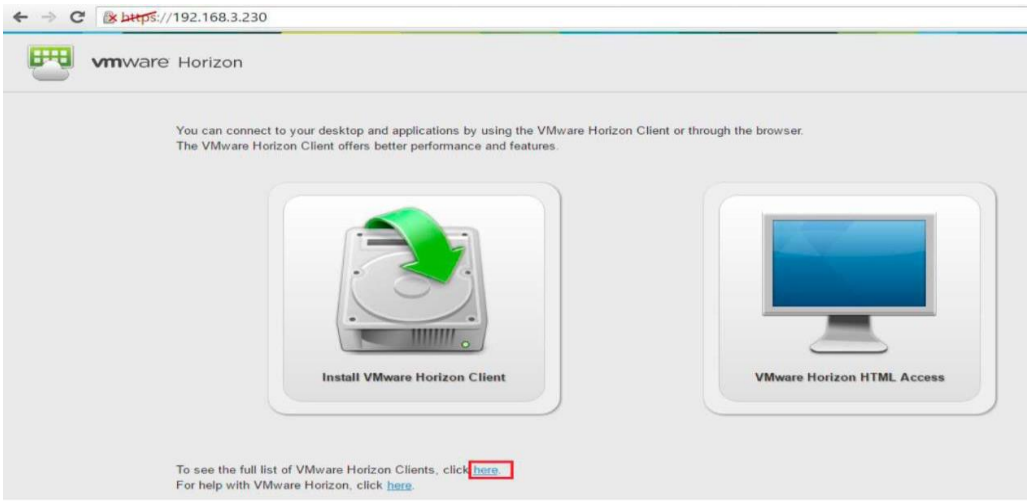


Рисунок 2.39 – Доступ до Connection Server через браузер

Установлення Horizon View Client на Windows. Скачуємо з Інтернету установник для Windows відповідно до розрядності ОС 32 або 64, запускаємо і вказуємо IP-адресу Connection Server (рисунок 2.40).

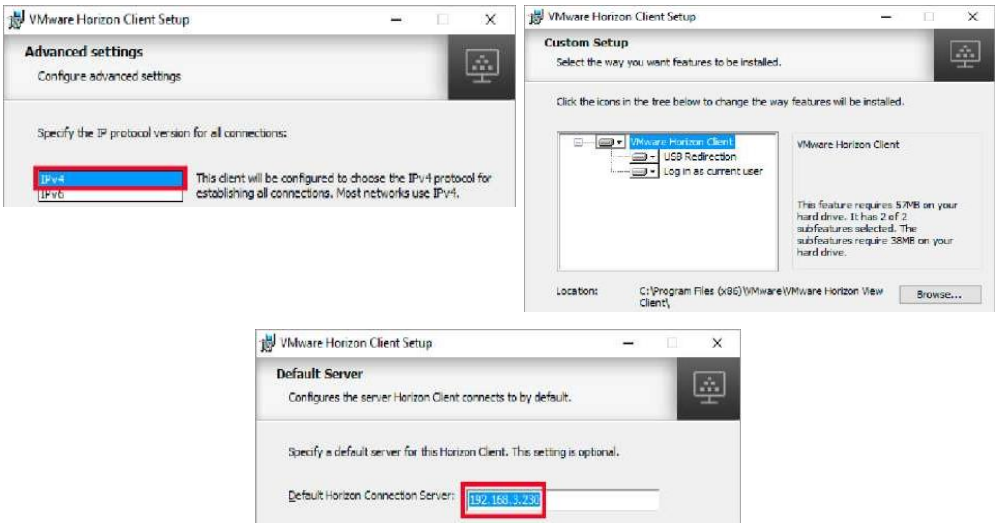


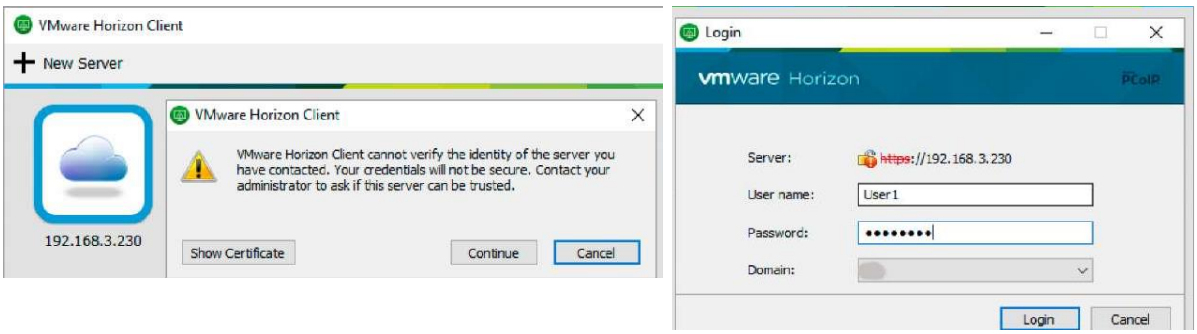
Рисунок 2.40 – Установлення Connection Server для Microsoft Windows

Якщо підключення буде здійснюватися з комп'ютера поза доменом domain.com (або ПК налаштований на інший DNS–сервер), то у файл C:\Windows\system32\drivers\etc\hosts необхідно додати 192.168.3.230 vconneci.domain. local.

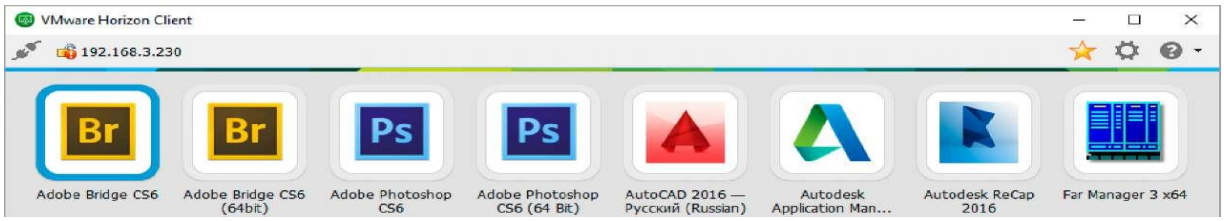
Запускаємо встановлений Клієнт, ігноруємо попередження про сертифікат після натискання Continue (на цьому етапі вони ще не згенеровані – повернемося до них пізніше), вказуємо логін і пароль користувача (рисунок 2.41, а), вибираємо і запускаємо додаток (рисунок 2.41, б).

Установлення Horizon View Client на Linux. Horizon View Client може бути так само встановлений на linux–дистрибутиви з перевіркою сумісності з будь-яким дистрибутивом і ознайомленням з докладними вимогами до системи. Зверніть увагу, що для функціонування кожного компонента Horizon View Client потрібна наявність у дистрибутиві певних бібліотек і створення до них лінків. Перевірити наявність бібліотек можна при установленні Horizon View Client, відповівши ствердно на запитання

[8–14]: Do you want to check your system compatibilities for Horizon Client, this Scan will NOT collect any of your data? [yes / no].



а)



б)

Рисунок 2.41 – Установлення Horizon View Client

Розглянемо процес установлення 64-розрядної VMware Horizon Client 4.3.0 на Elementary OS 0.3.2 Stable. Для цього дистрибутива необхідно додаткове створення лінків на libudev.so.0 (для RTAV і перенаправлення USB) і libffi.so.5 (для роботи протоколу PColP). Перевіримо наявність посилань на ці бібліотеки:

Ls –l/lib /x8б_б4–linux–gnu | grep libudev.so.0;

Ls –l/lib /x8б_б4–linux–gnu | grep libffi.so.5.

Якщо вивід обох команд порожній, то дивимося, на яку версію пакета належить створити посилання:

Ls –l / lib /x8б_б4–linux–gnu | grep libudev.so

lrwxrwxrwx 1 root root 1б Feb 25 201б libudev.so.1 –> libudev.so.1.3.5

–rw–r – r—1 root root б7б00 Feb 25 201б libudev.so.1.3.5.

Отже, створюємо посилання на libudev версії 1.3.5, підвищивши права до суперкористувача: Sudo su

Ln –sf /lib/x8б_б4–linux–gnu/libudev.so.1.3.5 /usr/lib/libudev.so.0

Аналогічно перевіряємо посилання для libffi:

Ls –l / lib /x86_64–linux–gnu | grep libffi.

Не виключено, що пакети не встановлені. Ставимо:

Apt–get install libffi6 libffi–dev *і створюємо софт–лінк:*

Ln –sf /usr/lib/x86_64–linux–gnu/libffi.so.6/usr/lib/libffi.so.5.

Для роботи Horizon View Client версії 4.0 і вище потрібно так само OpenSSL 1.0.2f або вище. Дивимося встановлену версію: Openssl version –*v* OpenSSL 1.0.1f 6 Jan 2014.

Встановлений пакет необхідно оновити, але в репозиторіях може не бути потрібної версії. Можна зібрати з початкових кодів. Спочатку перемикаємося на суперкористувача (якщо ще не перемкнулися) і створюємо папку, куди далі будемо завантажувати вихідні коди і встановлювати пакети: Sudo su

Mkdir / root / temp && cd / root / temp.

Спосіб 1 (через URL):

Apt–get install php5–curl

Apt–get install make

Curl https://www.openssl.org/source/openssl–L0.2g.tar.gz | tar xz # && cd openssl–1.0.2g && ./config && make depend && make && make install

Ln –sf / usr / local / ssl / bin / openssl[Л]which openssF

Openssl version —v.

Спосіб 2 (через wget).

Швидше за все пакет make вже встановлений в Elementary OS, тому в другому варіанті без його установлення:

Wget https://www.openssl.org/source/openssl–L0.2g.tar.gz – no-check – certificate

Tar –xzvf openssl–1.0.2g.tar.gz

Cd openssl–1.0.2g

#./config

Make depend

Make install

Ln –sf / usr / local / ssl / bin / openssl which openssL

Openssl version-v.

Створюємо софт-лінки для libssl і libcrypto:

Sudo ln–s/lib/x86_64–linux–gnu/libssl.so.1.0.0/lib/x86_64–linux–gnu/libssl.so.1.0.2

Sudo ln – s/lib/x86_64–linux–gnu/libcrypto.so.1.0.0/lib/x86_64–linux–gnu/libcrypto.so.1.0.2.

Приступаємо до процесу установлення Horizon View Client. Відкриваємо це посилання в браузері, копіюємо посилання для скачування на кнопці Download і ім'я викачуваного пакета. Потрібно виконати щось на зразок: Cd..Wget

https://download3.vmware.com/software/view/viewclients/CART16Q4/VMware – Horizon–Cl ient–4.3.0–4710754.x64. bundle

Chmod + x VMware–Horizon–Client–4.3.0–4710754.x64.bundle

./VMware–Horizon–Client–4.3.0–4710754.x64. bundle.

Якщо дзеркало для скачування виявиться недоступним протягом тривалого часу, то можна завантажити пакет вручну в браузері або залити його за допомогою WinSCP.

Читаємо ліцензійну угоду (погоджуємося – вводимо y), залишаємо ввімкненими всі опції (тиснемо Enter), реєструємо сервіс після установлення (відповідаємо y) і запускаємо тест сумісності, висновок якого буде таким:

Do you want to check your system compatibilities for Horizon Client,

this Scan will NOT collect any of your data? [yes / no]: y

Scanning libxml2.so.2 Please wait

[## ############] 100%

VMware Horizon Smart Card Success

VMware Horizon Real–Time Audio–Video Success

VMware Horizon Client Drive Redirection Success

VMware Horizon Multimedia Redirection (MMR) Success

VMware Horizon PCoIP

Success

VMware Horizon USB Redirection Success

VMware Horizon Virtual Printing Success

VMware Horizon Client Success.

Тест під час установлення має бути пройдений, в іншому випадку – можна переглянути висновок команди: Ldd/ usr / lib / vmware / view / bin / vmware–view на предмет рядків, що містять – not found.

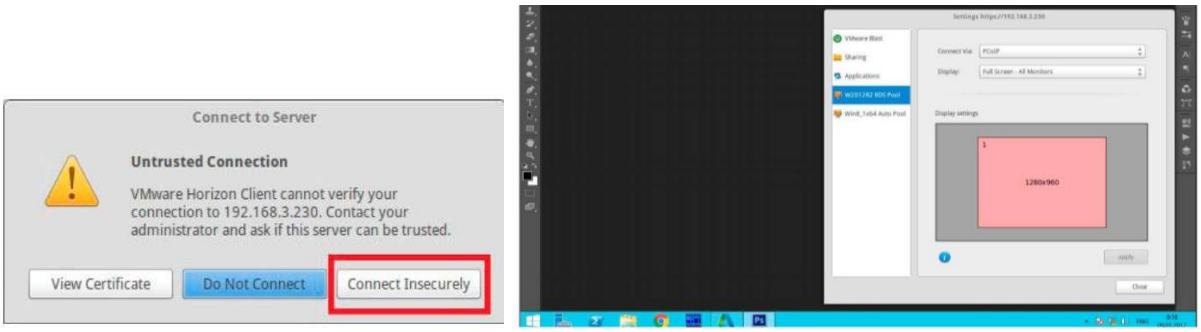
Якщо, наприклад, бачите:

libssl.so.1.0.2 => not found, libcrypto.so.1.0.2 => not found,

то це означає, що процес оновлення OpenSSL, описаний вище, не був проведений коректно або не створені софт-лінки. Задовольнити всі необхідні залежності методом установлення оновлення або створення лінків (для дистрибутивів, відмінних від Elementary, може знадобитися створення додаткових посилань) [14].

Тепер залишився фінальний штрих: скоріше за все ви плануєте підключення з машини, не введеної в domain.local. Тоді перед запуском Horizon View Client у файл hosts потрібно внести рядок, що містить IP-адресу та ім'я хоста Horizon Connection Server: Nano / etc / hosts 192.168.3.230 vconnect.domain. local.

Відкриваємо в лівому верхньому кутку Applications, запускаємо Horizon View Client, ігноруємо попередження, пов'язане з неправильним сертифікатом (рисунок 2.42, а), додаємо сервер (192.168.3.230) і перевіряємо працездатність клієнта з протоколами PColP і VMware Blast. Обидва протоколи виявилися цілком працездатними навіть при використанні Software Rendering (з увімкненою опцією Enable 3D graphics support і кількістю відеопам'яті рівною 128 Мбайт) під тестовою ВМ, розташованою на хості VMware ESXi (рисунок 2.42, б).

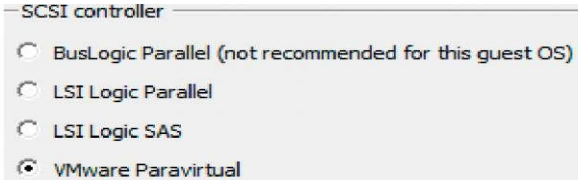


а) б)

Рисунок 2.42 – Horizon View Client под Linux

Підготовка ВМ. Створюємо ВМ (наприклад WIN7x64) і заходимо в її властивості. Для збільшення пропускної самоспроможності мережевих адаптерів вибираємо у властивостях віртуальної машини тип VMXNET 3 (рисунок 2.43, а) паравіртуалізований мережевий адаптер, спроєктований з розрахунком на максимальну продуктивність. Цей адаптер має пропускну спроможність 10 Gb/s, драйвери підтримки якого йдуть у складі VMware Tools [8–14].

а)



б)

Рисунок 2.43 – Створення Master Desktop Template

Наступним етапом у ролі контролера віртуальних жорстких дисків необхідно вибрати VMware Paravirtual SCSI (або PVSCSI) (рисунок 2.43, б). PVSCSI дає більшу продуктивність, ніж звичайний LSI-контролер, але тут

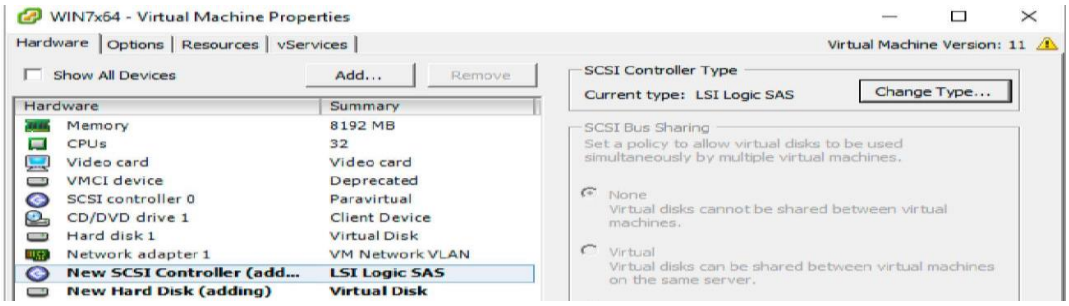
є одне зауваження: деякі ОС, такі як, наприклад, Windows 7, не встановлюються на жорсткий диск, підключений до PVSCSI-контролера. Для того щоб робота PVSCSI в Windows 7 була можливою, необхідно створити Master Desktop Template (рисунок 2.44, а, б).

2.3 Порядок виконання лобораторної роботи

1 Встановити Windows, використовуючи дефолтний контролер (LSI Logic SAS). Встановити VMware Tools.

2 Додати додатковий жорсткий диск на іншу SCSI-шину: наприклад, якщо основний жорсткий диск позначено як SCSI (0: 0), то додатковий буде SCSI (1: 0) (рисунок 2.44, а). При додаванні пристрою на окрему SCSI-шину буде так само доданий додатковий контролер, тип якого потрібно змінити на VMware Paravirtual (рисунок 2.44, б).

а)



б)

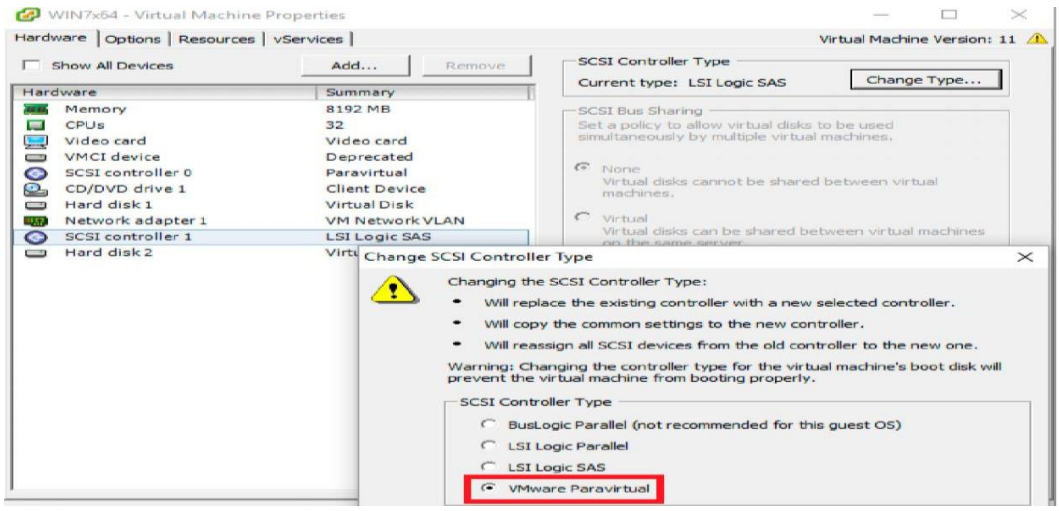


Рисунок 2.44 – Створення Master Desktop Template

3 Завантажуємо ВМ, відкриваємо диспетчер пристроїв, де має з'явитися VMware Virtual disk SCSI Disk Device (рисунок 2.45).

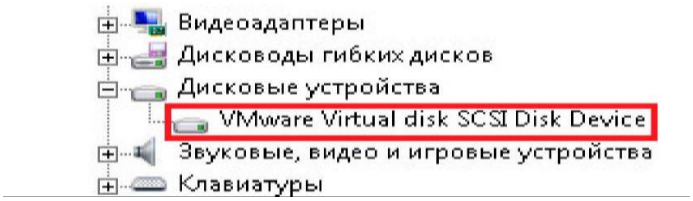


Рисунок 2.45 – Створення Master Desktop Template

4 Вимикаємо ВМ, видаляємо додатковий жорсткий диск і міняємо тип основного SCSI-контролера на VMware Paravirtual. Тепер система має завантажитися. Далі необхідно ввімкнути Memory і CPU Hot Add (рисунок 2.46) і видалити Floppy drive у конфігурації ВМ. Видалення Floppy drive даст змогу заощадити значну кількість ресурсів, особливо якщо ви збираєтеся створювати кілька десятків клонів чи комусь із користувачів знадобиться віртуальний Floppy drive.

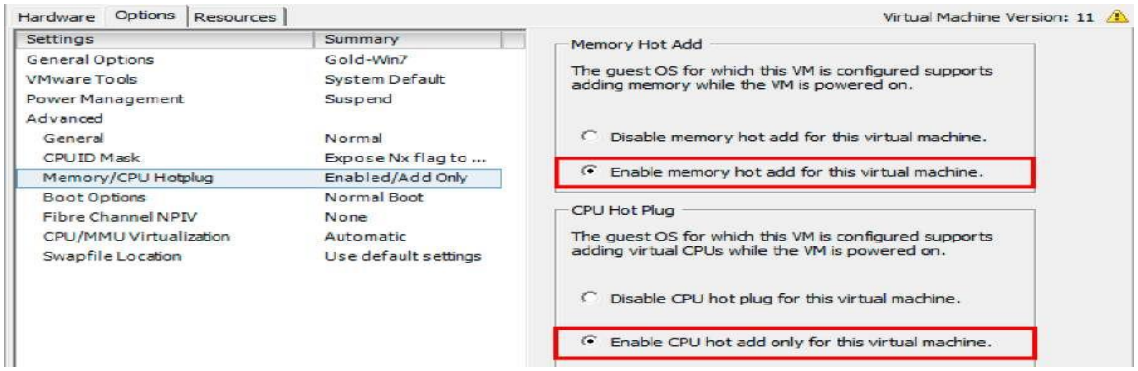


Рисунок 2.46 – Створення Master Desktop Template

Відкриваємо у властивості віртуальної машини вкладку Options і виставляємо: Boot Options: Boot to BIOS.

Заходимо в BIOS і відключаємо Floppy drive (рисунок 2.47).

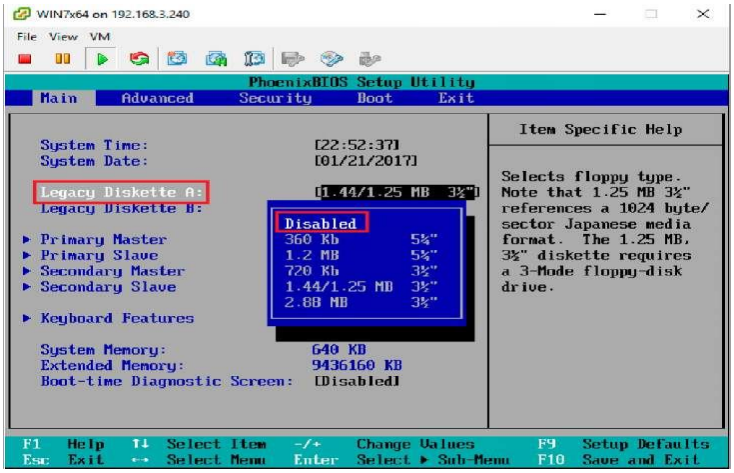


Рисунок 2.47 – Створення Master Desktop Template

Оптимізація ОС і створення шаблона. Усі дії з налаштування гостьової ОС необхідно виконувати з консолі ESXi (не через Remote Desktop). Перед оптимізацією системи слід встановити VMware Horizon Agent і обов'язково вибрати компонент VMware Horizon Instant Clone Agent (рисунок 2.48).

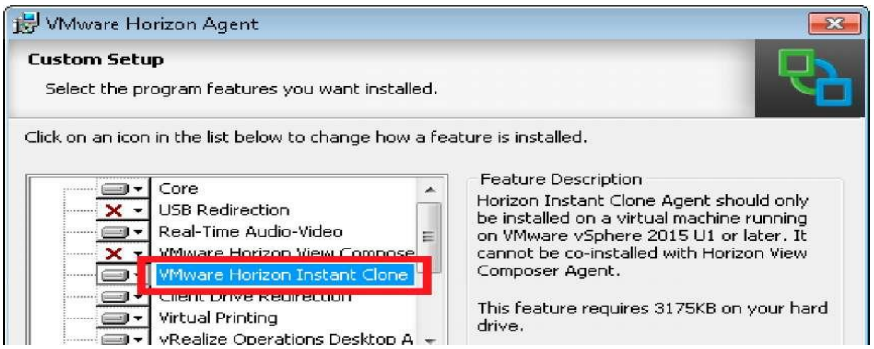


Рисунок 2.48 – Вибир компонента VMware Horizon Instant Clone Agent

Тепер можна приступити до оптимізації ОС, але перш ніж завантажити утиліту VMware OS Optimization Tool, переконайтеся, що встановлено NET Framework 3.5. Застосуйте всі необхідні налаштування і після завершення натисніть кнопку Optimize (рисунок 2.49). Тепер можна встановити все необхідне ПЗ: FoxIt Reader, LibreOffice і т. д.

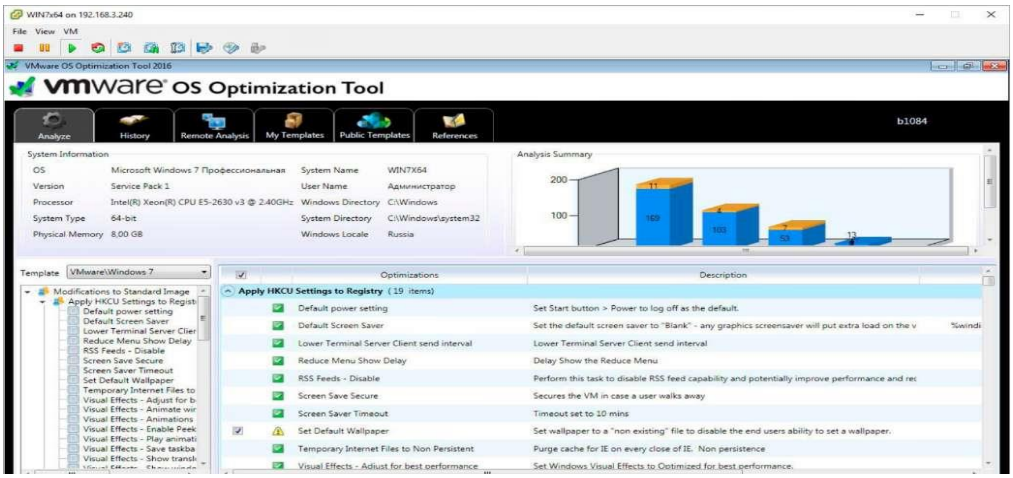


Рисунок 2.49 – VMware OS Optimization Tool

Тепер, коли всі програми встановлені, переходимо в vSphere Client, підключаємося до vCenter Server (у нашому випадку – 192.168.3.243) і правою кнопкою на тільки-но налаштованій ВМ (WIN7x64) вибираємо: Template –> Clone to template, де в режимі діалогу задаємо ім'я створюваного template ВМ, вибираємо кластер, хост і datastore, примусово задавши Thin Provision для економії дискового простору (рисунок 2.50).

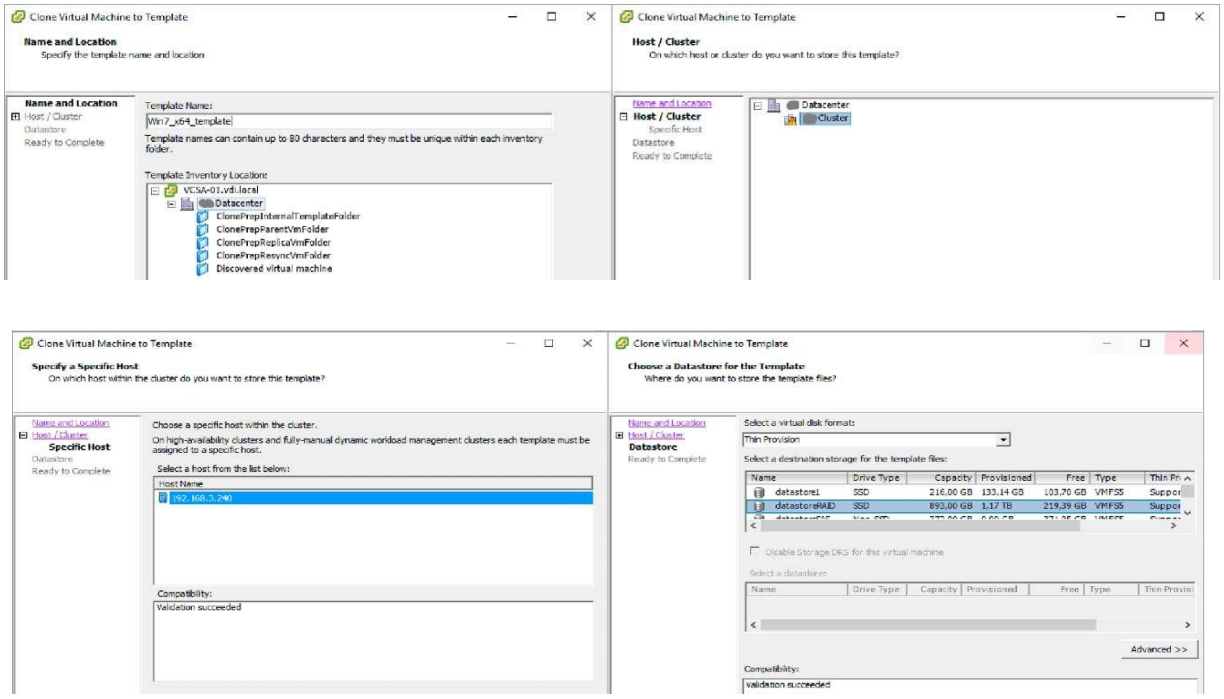


Рисунок 2.50 – Створення template ВМ

Не закриваючи все той же vSphere Client, переходимо:

Home –> Customization Specifications Manager –> New

і створюємо новий шаблон кастомізації гостьових систем. Далі в режимі діалогу потрібно буде задати параметри гостьової системи (рисунки 2.51, 2.52).

Для створення унікального NetBIOS для кожної ВМ ставимо чек-бокс на опції Append a numeric value to ensure uniqueness (рисунок 2.53), а як налаштування мережі вибираємо Typical settings (отримання IP-адреси за DHCP).

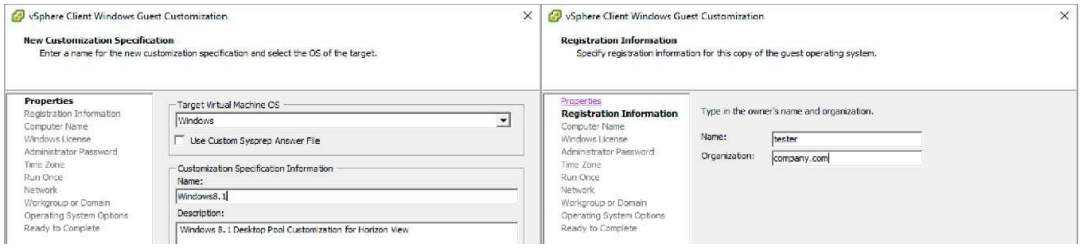


Рисунок 2.51 – Guest Customization

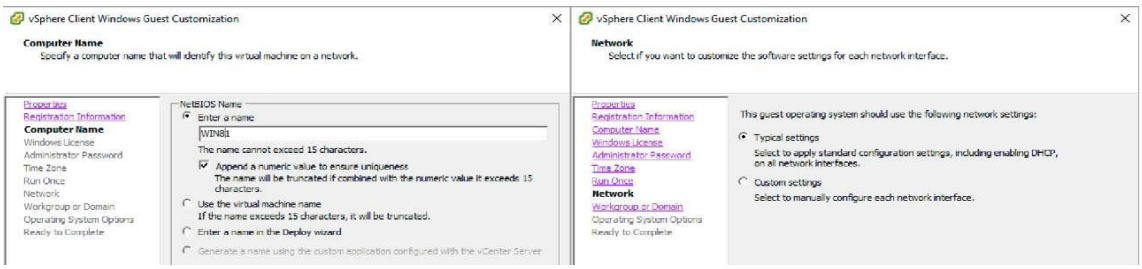


Рисунок 2.52 – Guest Customization

Рисунок 2.53 – Guest Customization

Створення Automated Desktop Pool. Для створення автоматичного пулу десктопів необхідно відкрити в браузері консоль адміністрування Horizon, перейти в Inventory –> Catalog –> Desktop Pools і, натиснувши кнопку Add, перейти до діалогу створення пулу десктопів (рисунок 2.54).

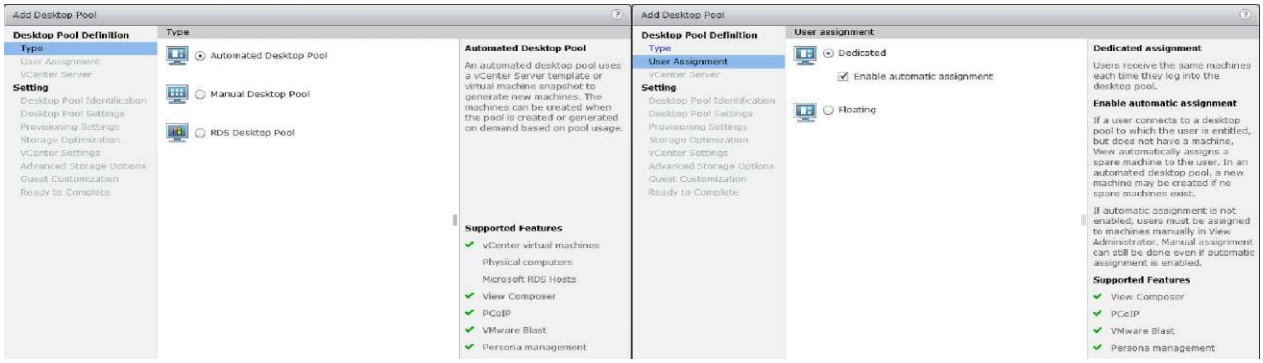


Рисунок 2.54 – Створення Automated Desktop Pool

При створенні автоматичного пулу необхідно вибрати виділену віртуальну машину (Dedicated Assignment) і встановити чек-бокс на автоматичному призначенні (Enable automatic assignment) [8–14].

Увімкнення опції Automatic assignment дає змогу автоматично призначати користувачу віртуальну машину з заданого пулу. Якщо користувач підключається до пулу, до якого має право, але віртуальної машини не має, то Horizon View присвоїть вільну віртуальну машину. Якщо в автоматизованому пулі не існує вільних віртуальних машин, то буде створена нова.

Якщо опція Automatic assignment не вибрана, віртуальні машини для підключення мають бути призначені через консоль адміністрування Horizon View вручну. Примусове призначення віртуальних машин користувачам так само можливо, якщо ввімкнена опція Automatic assignment.

Створимо повні клони віртуальних машин (поки що без використання Composer Server), вибираємо Full Virtual Machines. Зверніть увагу, що кожна наступна клонована віртуальна машина буде займати стільки ж місця, скільки і Master Desktop Template.

Далі потрібно задати ім'я пулу (ID), що відображується для користувача, у View Client ім'я пулу (Display Name) і групу, якій дозволено доступ до цього пулу (рисунок 2.55).

Далі зібрані всі налаштування пулу (рисунок 2.56): заборонні правила для підключення (Connection Server restrictions), відключення користувача після закінчення часу (Automatically logoff after disconnect), дозвіл користувачам скидати віртуальні машини (Allow users to reset their machines).

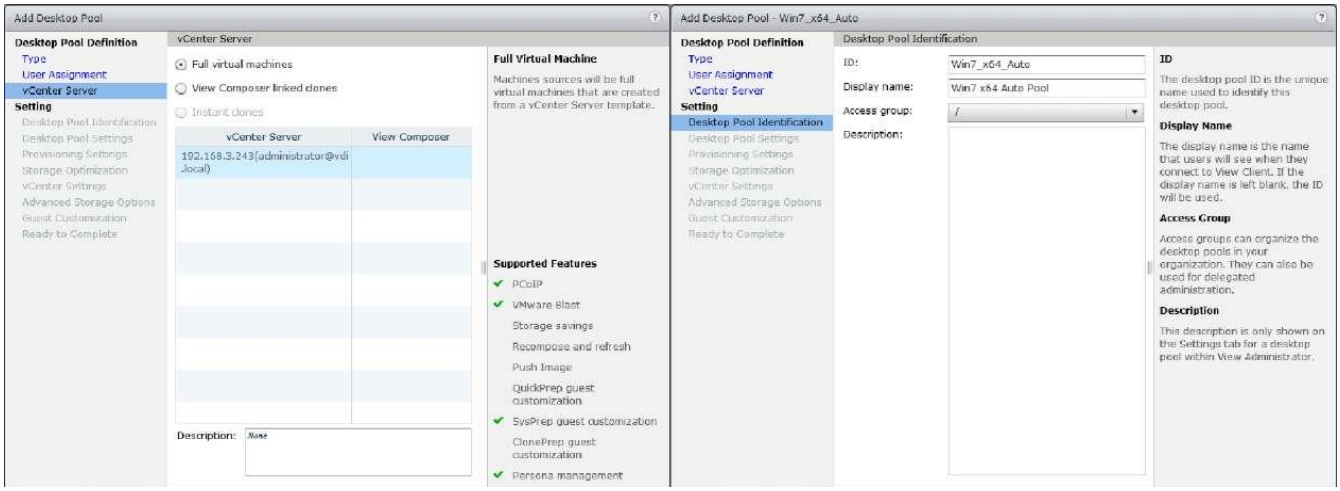


Рисунок 2.55 – Створення Automated Desktop Pool

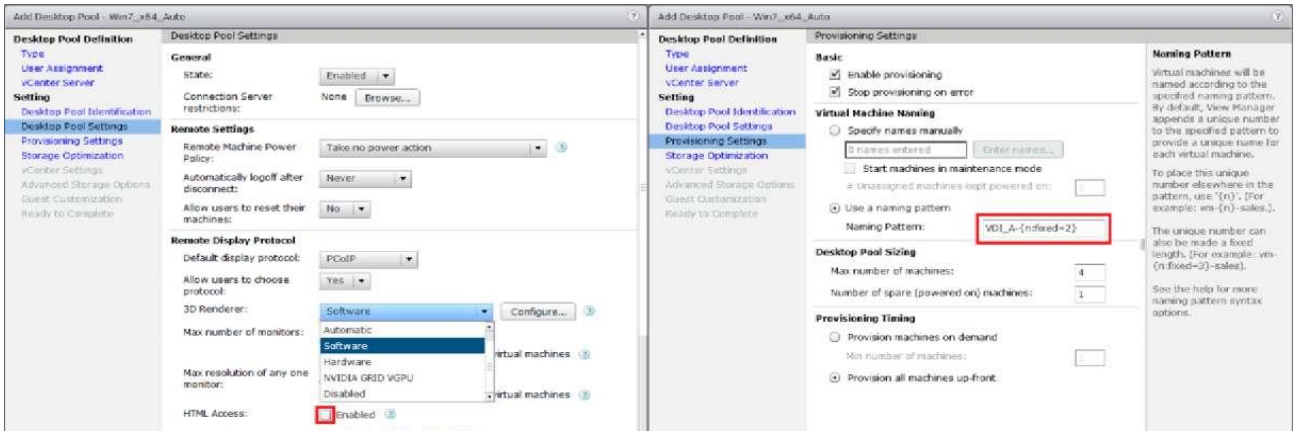


Рисунок 2.56 – Створення Automated Desktop Pool

Нижче розташовані налаштування протоколу та акселерації: протокол за умовчанням (Default display protocol), можливість вибору протоколу користувачами (Allow users to choose protocol), 3D-рендеринг (і регулювання кількості використовуваної відеопам'яті), максимальна кількість моніторів, максимальний дозвіл на один монітор і доступ через HTML, який найкраще ввімкнути [8–14].

Опції Adobe Flash можна залишити за замовчуванням: не контролювати якість Adobe Flash і Adobe Flash throttling, що дає змогу регулювати кількість кадрів за секунду на основі агресивності обраного параметра. Ми залишаємо в положенні disabled. Аналогічно знімаємо чек-бокс у настройках Mirage Server, який поки що не встановлено.

На наступному екрані (Provisioning settings) залишаємо обидва чек-бокси на Enable Provisioning і зупиняємо розміщення віртуальної машини при помилці (Stop provisioning on error). Як патерн для присвоєння імен віртуальних машин задаємо вираз VDI A– {n: fixed = 2}.

Отже, створені віртуальні машини Automated Desktop Pool будуть названі VDI–A–01, VDI–A–02 тощо.

У полі Desktop Pool Sizing задаємо максимальну кількість увімкнуних машин, а в полі Provisioning Timing встановлюємо чек-бокс на Provisioning all machines up–front, що передбачає створення клонів на етапі розгортання пулу, а не на вимогу (on demand) [8–14].

У налаштуваннях Storage Policy Management (рисунок 2.57) можна поки не використовувати VMware Virtual SAN (якщо Virtual SAN ще не встановлений, то налаштування будуть неактивними) і переходимо до вибору вже створеного вище темплейта ВМ (Virtual Machine Template), а також визначення місця розташування ВМ (папка, хост або кластер, пул ресурсів і datastore) [8–14].

Переходимо до Advanced Storage Options (рисунок 2.58). Починаючи з vSphere 5.x можна використовувати View Storage Accelerator, що дає змогу збільшити продуктивність за рахунок кешування даних пулу десктопів. За бажанням можна так само налаштувати Blackout Times –

розклад (дні тижня і часові інтервали), протягом якого оновлення кешу проводитися не буде.

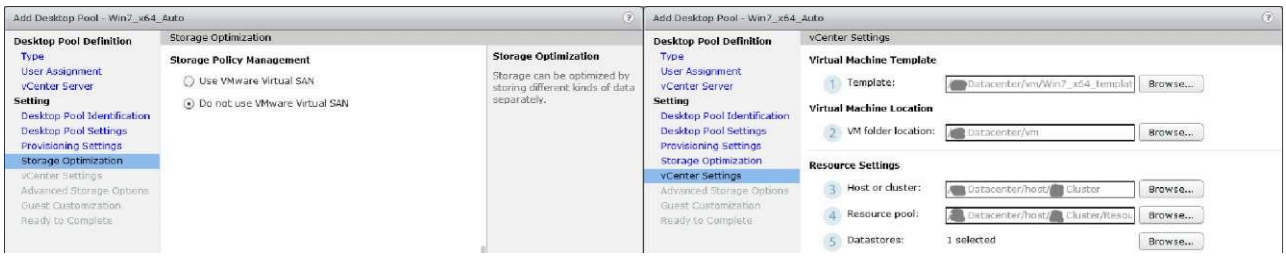


Рисунок 2.57 – Створення Automated Desktop Pool

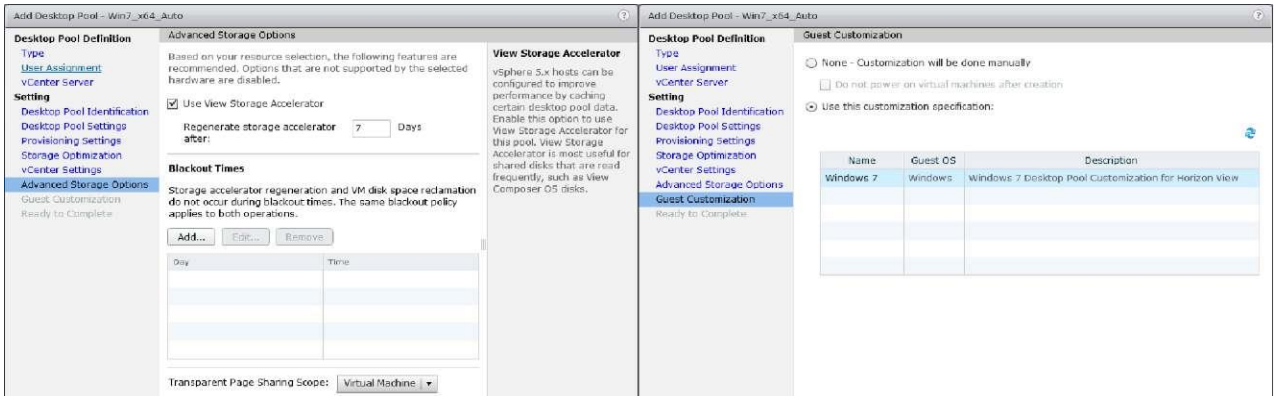


Рисунок 2.58 – Створення Automated Desktop Pool

Ми залишаємо значення за замовчуванням для Regenerate storage accelerator after (сім днів) і переходимо до фінальних налаштувань пулу – Guest Customization, де вибираємо створений раніше шаблон кастомізації операційної системи: Windows 7 Desktop Pool Customization For Horizon View. Звіряємо всі введені значення на екрані Ready to complite, ставимо чек-бокс на опції Entitle users after wizards finishes і тиснемо кнопку Finish. У вікні призначаємо користувачів для доступу до створюваного пулу десктопів [8–14].

Тепер, зайшовши в vSphere Client, бачимо, що почався процес клонування ВМ (рисунок 2.59).

Так само у вебконсолі адміністрування Horizon за подвійним натисканням на створюваний пул (меню *Catalog –> Desktop Pools*) у вкладці Inventory можна побачити статус Provisioning (рисунок 2.60), а через деякий час – Customizing для кожної створюваної віртуальної машини пулу. Тут же при створенні пулу можна побачити повідомлення про помилки, наприклад при нестачі вільного місця на datastore: Error during provisioning: *No suitable datastores* або помилку вигляду *No users or groups are entitled to this pool*.

Рисунок 2.59 – Процесс клонування ВМ для Automated Desktop Pool

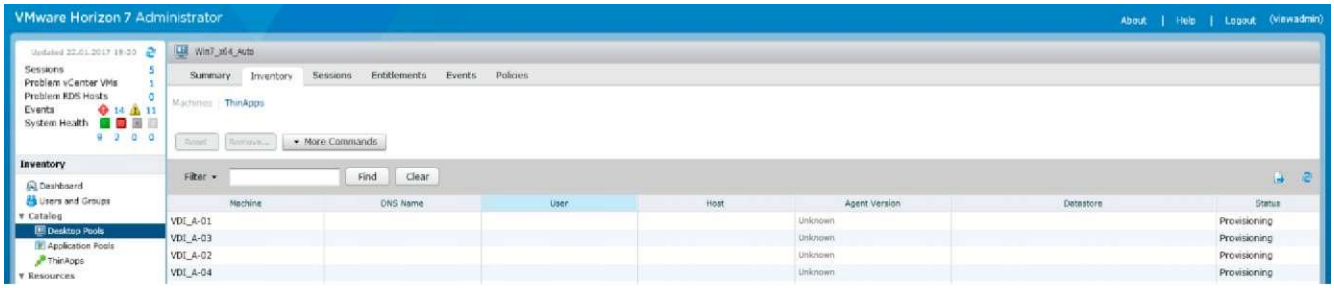


Рисунок 2.60 – Статус Provisioning

Натиснути кнопку *Click Entitlements ... to add users to this pool*, яка означає, що були додані користувачі для доступу до створюваного пулу, процес додавання яких буде розглянуто нижче. Тут же, вибравши Entitlements або перейшовши в *Catalog –> Desktop Pools –> Entitlements ... –> Add entitlment...*, можна перепризначити користувачів для доступу до створюваного пулу десктопів (рисунок 2.61).

У вкладці Policies можна ввімкнути перенаправлення мультимедіа-контенту на бік клієнта, проте слід переконатися, що використовуваних ресурсів буде достатньо для декодування. Можна так само

використовувати USB- пристрої, підключені до клієнтів, виставивши значення Allow в USB acess, а при ввімкненні апаратної акселерації протоколу PCoIP (PCoIP hardware acceleration) можна так само вибрати пріоритет [8–14].

Після завершення процесу клонування (рисунок 2.62) можна підключитися до пулу десктопів через View Client (рисунок 2.63), а статус підключення можна перевірити в консолі адміністратора Horizon, вибравши пул десктопів і перейшовши у вкладку Sessions.

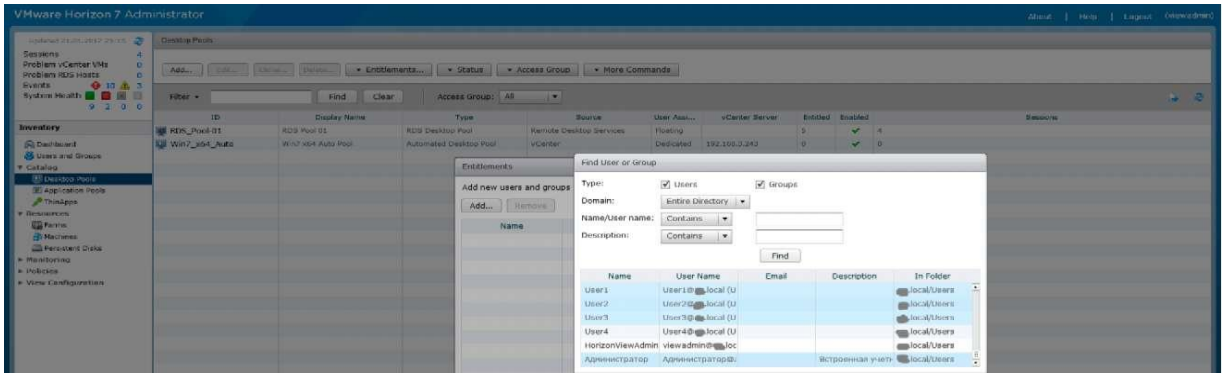


Рисунок 2.61 – Призначення користувачів для доступу до Desktop Pool

Рисунок 2.62 – Завершення процесу клонування ВМ
для Automated Desktop Pool

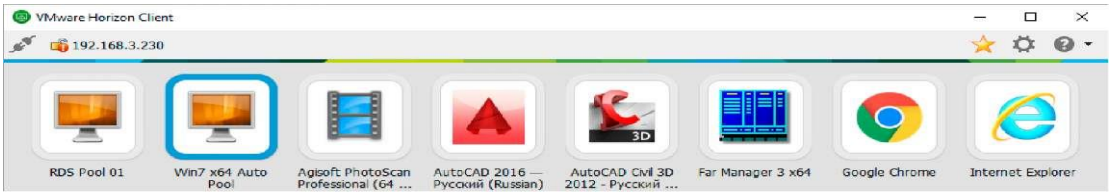


Рисунок 2.63 –Підключення до Automated Desktop Pool

Отже, при використанні Automated Desktop Pool, на відміну від ферми RDS, кожен користувач підключається до виділеної (dedicated) віртуальної машини. Automated Desktop Pool і повне клонування віртуальних машин відносно прості в розгортанні, тому що не потребують розгортання окремих компонентів, але очевидним недоліком повного клонування віртуальних машин є спожитий дисковий простір і серйозні вимоги до пропускної спроможності дискових накопичувачів.

2.4 Зміст звіту

Звіт оформляється кожним здобувачем індивідуально і має містити:

– титульний аркуш з номером і назвою та метою роботи;

– скріншоти з виконання роботи;

– результати виконання роботи та висновки з роботи.

Контрольні запитання

1 Що являє собою рішення VMware Horizont?

2 Які основні особливості програмного рішення з віртуалізації VMware Horizont?

3 Які платформи для побудови системи віртуалізації вам відомі?

4 У чому особливості роботи з різними системами віртуалізації?

5 Які переваги застосування віртуалізованих серверних рішень?

6 Які інформаційні технології використовують для побудови системи віртуалізації ОС?

Лабораторна робота 3. Ознайомлення з платформою хмарних обчислень GOOGLE APP ENGINE

Мета роботи - навчитися виконувати налаштування комплексних рішень з побудови віртуальних інфраструктур організацій, у тому числі з віддаленими офісами та особливим доступом до їхніх мереж.

3.1 Методичні вказівки з організації самостійної роботи здобувачів

Почнемо з того, що існує безліч видів віртуалізації, але в рамках цього матеріалу будуть названі три. Перший – це віртуалізація подання. Найяскравіший приклад – термінальні служби Windows Server. У цьому випадку клієнтський додаток виконується на сервері, а клієнт бачить тільки інтерфейс. Така модель доступу дає змогу знизити вимоги до апаратного забезпечення на боці клієнта і підвищити безпеку [1–7].

Що стосується обладнання, то як термінальні клієнти можуть використовуватися навіть смартфони. Це може стати наступним великим кроком у корпоративному середовищі. Уже сьогодні існують постачальники, які пропонують віртуальну мобільну інфраструктуру (VMI). Тут мобільний телефон користувача виступає в ролі «тонкого клієнта», а ОС працює на віддаленому сервері. Таке рішення вже набуває популярності у прихильників концепції BYOD (Bring Your Own Device).

Другий вид віртуалізації – віртуалізація додатків. Віртуалізація додатків дає змогу запускати окремий додаток в ізольованому середовищі, так званій «пісочниці». Отже, воно не здатне завдати шкоди операційній системі та іншим програмам, що дає можливість запускати на одному комп'ютері кілька програм, які конфліктують одна з одною [1–7].

Третій вид – це віртуалізація серверів, імітація апаратного забезпечення комп'ютера за допомогою спеціального програмного забезпечення. На такий віртуальний комп'ютер можна встановити ОС, і вона буде працювати так само, як на фізичній машині. Найцікавіша перевага цієї технології – це можливість запуску декількох віртуальних

комп'ютерів усередині одного «залізного», що дає змогу максимально задіяти ресурси фізичного сервера.

Але де можна застосовувати віртуалізацію і яка з цього вигода? Віртуалізацію можна використовувати для тестування нових рішень, а також навчання персоналу. Наприклад, можна протестувати розроблений додаток або «латочку» до нього перед запуском у production. Щоб зробити це, не доведеться докуповувати обладнання і ризикувати збоєм роботи всієї IT-інфраструктури в разі помилки [1–7].

Але віртуалізація серверів прийнятна і для роботи в самих GOOGLE APP ENGINE-середовищах. Тому є кілька причин. Перш за все це економія коштів. Завдяки консолідації ця технологія дає можливість скоротити кількість серверів. Якщо раніше для виконання якогось завдання було потрібно кілька фізичних машин, то тепер можна запустити потрібну кількість гостьових ОС у віртуальному середовищі на одному сервері. Це дає змогу економити на підтримці обладнання.

Одночасно з цим з'являється можливість скоротити кількість обслуговуючого персоналу. Зрозуміло, хтось, як і раніше, має стежити за здоров'ям хост-серверів, але якщо парк машин скоротився з 200 до 50 примірників, то можна зменшити і кількість фахівців.

Зрозуміло, у медалі є і зворотний бік. При впровадженні рішення на базі віртуалізації доведеться купувати нове «залізо». Справа в тому, що віртуальні машини досить «ненажерливі» і потребують багато ресурсів для роботи. Відповідно знадобляться більш потужні процесори, додаткові об’єми пам'яті, швидші системи зберігання.

З першої причини випливає друга: зменшення кількості фізичних серверів веде до зменшення займаного ними місця. Завдяки цьому знижується орендна плата за стійки в дата-центрі (ДЦ). Якщо компанія має власний центр опрацювання даних (ЦОД) або центр зберігання та обробки даних (ЦЗОД), то це означає загальне зниження енергоспоживання і тепловиділення системи, отже, з'являється можливість закуповувати менш потужні і дешевші системи охолодження, що неодмінно позначиться в рахунках на електрику.

Також слід зазначити, що віртуалізація дає змогу знизити витрати на адміністрування інфраструктури. Однією з головних переваг є можливість

віддаленого доступу до консолі управління. Так, щоб перезавантажити сервер, більше не потрібно бігати в машинний зал або використовувати перемикачі на базі IP, достатньо подати команду перезавантаження через консоль. Так само можна підключити додаткові обчислювальні ресурси [1–7].

Крім того, ще одним плюсом віртуалізації є простота клонування віртуальних машин. Припустимо, що серверна інфраструктура компанії стандартизована і являє собою групу серверів з однаковими налаштуваннями. Якщо керівництвом організації буде прийнято рішення відкрити додатковий офіс, то розгортання нової інфраструктури зведеться до копіювання образів і налаштування ПЗ.

Отже, перехід на віртуальну інфраструктуру дає можливість спрямувати всі свої зусилля на поліпшення якості сервісу. Але незважаючи на те, що віртуалізовані і хмарні середовища відкрили перед бізнесами нові шляхи розвитку, вони також принесли з собою нові ускладнення.

Звичайно, у віртуалізації є і свій недолік – необхідність перебудови підходу до роботи з надійністю системи. Дійсно, оскільки на одному фізичному сервері одночасно запущені кілька віртуальних машин, то вихід з ладу хоста призводить до одночасної відмови всіх обчислювальних машин (ОМ) і додатків, що працюють на них. Тому доцільно використовувати відмовостійкі рішення, наприклад на базі відмов кластерів.

У цьому випадку два або кілька серверів працюють у групі і для користувача виглядають як один сервер, який обробляє запити і відповідає за роботу додатків. У разі виходу з ладу одного з вузлів кластера призначені для користувача програми виконують failover. Відбувається автоматичний перезапуск на працездатних вузлах, а додаток або не припиняє роботу, або припиняє на досить короткий час [1–7].

Ще одне питання – балансування навантаження. Наприклад, якщо ВМ використовує багато обчислювальних ресурсів процесора (або пам'яті), то це позначається на роботі інших ВМ хоста, яким також потрібен процесорний час (пам'ять). Адміністраторам доводиться розподіляти навантаження, встановлюючи правила, за якими запущені віртуальні машини будуть автоматично переміщатися на менш навантажені сервери або ж розвантажувати завантажені.

Технології віртуалізації покликані вирішити питання з недостатнім ступенем утилізації ресурсів фізичних систем, але тепер з'явилася інша небезпека. Люди почали створювати ВМ при кожному зручному випадку, часто забуваючи видаляти старі і невикористовувані машини. Це веде до ускладнень з нестачею дискового простору, підвищеного споживання обчислювальних ресурсів і перебоїв у роботі.

Дуже часто керівництво компаній вважає віртуалізацію панацеєю, що веде до ще більшого витрачання ресурсів. Слід пам'ятати, що віртуальні технології – це просто інструмент, що має свої переваги і недоліки [1–7].

На сьогодні проєкти з віртуалізації IT-інфраструктури активно впроваджуються багатьма провідними компаніями. Вендори різних платформ віртуалізації можуть навести приклад успішних рішень, впроваджених у великих банках, державних і освітніх установах, навіть лікарнях. Усі ці компанії користуються перевагами технології і економлять на обслуговуванні, персоналі, апаратному забезпеченні (АЗ).

Дуже часто віртуалізацію обговорюють у контексті великих компаній, тому що їхні ресурси дають можливість отримати максимум з віртуальної IT-інфраструктури. Однак не слід забувати і про маленькі організації, які теж здатні отримати свою частку вигоди.

Оскільки обчислювальні потужності продовжують зростати, часто нема сенсу мати безліч серверів, коли один або декілька з них прекрасно впораються з необхідними завданнями. Однак консолідація серверів для деяких бізнесів не є необхідністю. Для компаній з простими і невеликими IT-інфраструктурами віртуалізація може виявитися не найкращим рішенням.

Якщо в компанії один додаток і лише два сервери, то перехід на віртуальну інфраструктуру може не принести очікуваного зиску, як каже Шон Селлерс (Shaun Sellers), продукт-менеджер Vision Solutions. Він також додає, що компаніям, у яких працюють лише два співробітники, слід звернути увагу на хмарні сервіси замість віртуальних технологій [1–7].

Ще одна переваге віртуалізації – вона дає можливість автоматизувати трудомісткі завдання, на які витрачається багато часу IT-персоналу. Віртуалізовані платформи дають змогу робити більше при

менших витратах і легко масштабуються зі зростанням запитів, як зауважує Джейсон Бейтер (Jason Beiter), архітектор корпоративних рішень з Annese & Associates Inc.

Наприклад, якщо компанії потрібно додати новий сервер, то при традиційній інфраструктурі будуть потрібні значні фінансові вливання і часові витрати. При віртуалізації це робиться практично миттєво. Отже, якщо ви очікуєте найближчим часом значне зростання і бурхливий розвиток компанії, то, можливо, має сенс перейти на віртуальні технології.

Оскільки більшість систем віртуалізації орієнтовані на великі компанії і корпорації, маленьким бізнесам може бути непросто знайти те, що задовольнить їхні потреби. Багато провайдерів, навіть таких великих, як VMware, Citrix і Microsoft, позиціонують себе як small-business-friendly [1–7].

3.2 Практична частина

3.2.1 Установлення і розгортання View

У попередній лабораторній роботі було розпочато розгортання основних компонентів VMware Horizon View, тепер розглянемо вбудовані технології клонування десктопів і безпечне підключення до інфраструктури View.

Ще влітку 2014 р. VMware анонсувала технологію VMware Project Fargo (початкова назва – VM Fork) (рисунок 3.1), а з виходом сьомої версії Horizon і гіпервізора ESXi v6 Update 1 на зміну пов'язаним клонам (Linked-Clones) з'явилася технологія миттєвих клонів (Instant Clones), що дає можливість створити desktop ще швидше (ніж було раніше з використанням Linked-Clones) за рахунок кількості стадій клонування віртуальної машини [8–14].

Суть технології VM Fork в тому, що батьківська віртуальна машина на деякий час заморожується і створюється дочірній клон, або відросток (fork), який починає використовувати ту саму оперативну пам'ять (Shared Memory), але при цьому всі зміни дочірньої віртуальної машини зберігаються окремо від батьківської. Аналогічно відбувається і з

жорстким диском: усі наступні зміни щодо батьківської віртуальної машини пишуться на Delta Disk. Після заморожування батьківської ВМ за допомогою технології Fast Suspend Resume (FSR) і створення клону (або відростка) відбувається реконфігурація ВМ: пристрої отримують новий UUID, мережевий адаптер отримує нову MAC-адресу тощо.

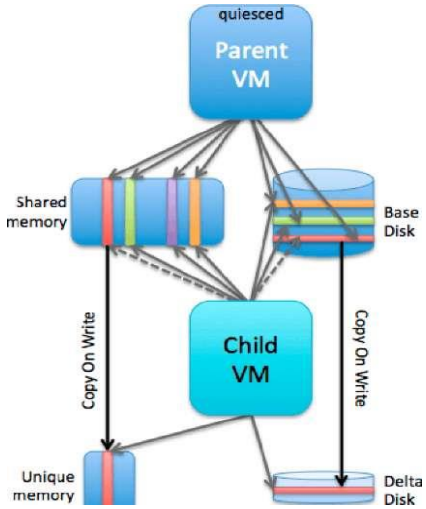


Рисунок 3.1 – Принцип роботи VMware Project Fargo

Що стосується вже звичної технології клонування, то для її використання буде потрібна окрема ВМ з встановленим Composer ServerW:

– vcomposer.domain.local (192.168.3.232) – Horizon View Composer Server;

– при створенні сертифікатів для SSL-з'єднань буде потрібна окрема від доменного контролера ВМ з розгорнутим Центром Сертифікації Active Directory [8–14]:

– pki–01.domain.local (192.168.3.237) – Active Directory Certificate Services.

3.2.2 Створення Instant Clone Desktop Pool

Instant Clone – технологія in–time доставки десктопів, що дає можливість швидкого розгортання дочірніх віртуальних машин з батьківського образу ВМ. Instant Clone з'явилися в Horizon 7 і використовують технологію Project Fargo (або vmFork), що стала

доступною в ESXi Update 1. Instant Clone дають змогу створити повнофункціональний десктоп за 1.5–2 с, а так само використовувати менше дискового простору, ніж повні клони ВМ [8–14].

Вимоги:

– ліцензія Horizon редакції Enterprise;

– ВМ Master Dekstop Template в.11 (VM Hardware V. 11);

– можливо клонувати Windows 7, Windows 10 (Long-Term Servicing Branch (LTSB) Version 1507 (RTM), Version 1607 і Current Business Branch (CBB) Version 1511) і Windows Server 2016. Windows 8 / 8.1 зараз не підтримується. При цьому не можна оновлювати Master Desktop Template з Windows 8.1 до Windows 10 – тільки Windows 10, встановлену з нуля;

– KMS-сервер для активації Windows;

– на гостьову ОС Master Desktop Template необхідно встановити Horizon View Agent з увімкнутою опцією VMware Horizon Instant Clone (рисунок 3.2);

– хости з ESXi 6.0 Update 1 або вище;

– увімкнений View Storage Accelerator (рисунок 3.3);

– DHCP-сервер.

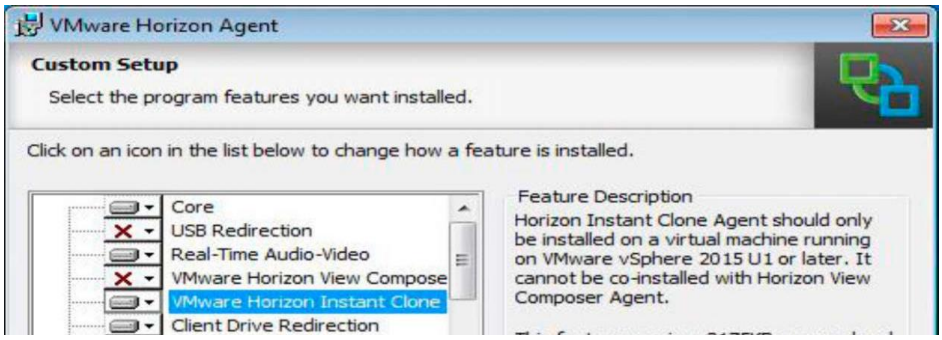


Рисунок 3.2 – Увімкнення опції VMware Hrozion Instant Clone в Horizon Agent

Для створення Instant Clone Desktop Pool як гостьової ОС будемо використовувати Windows 7 Professional (процес створення ВМ Win7x64 як Master Desktop Template був розписаний у попередній частині). Аналогічно рекомендується використовувати як контролер жорстких дисків VMware

Paravirtual, а як мережевий - адаптер VMXNET3. Так само рекомендується встановити hotfix433809 для Windows 7 SP1, щоб уникнути помилки при клонуванні [8–14]: After waiting for 600 seconds internal template VM: <VM name> is still has not finished customization. Giving up.

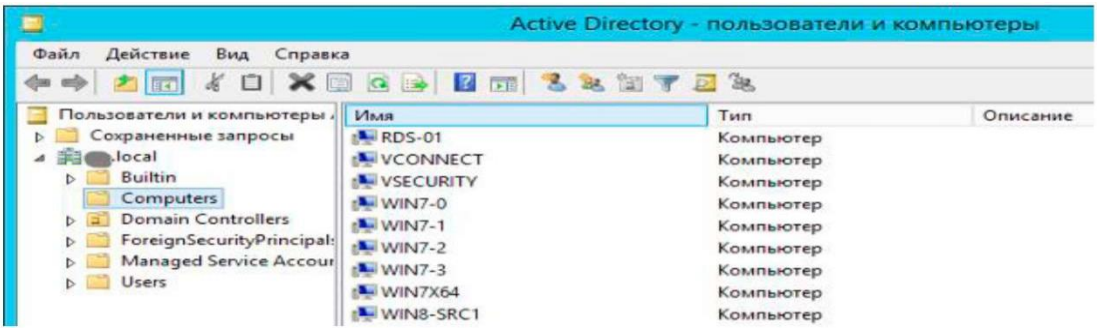


Рисунок 3.3 – Active Directory-користувачі і комп'ютери

Перед вимиканням ВМ переконаємося, що операційна система з'єднана з доменом domain.local, а в налаштуваннях мережі стоїть отримання адреси за DHCP.

Вимикаємо ВМ і створюємо snapshot (наприклад назвемо його Instant Clone Master). Зверніть увагу, що snapshot в vSphere зберігає так само і конфігурацію, тому уважно поставтеся до конфігурації віртуальної машини. Наприклад, встановіть кількість vCPU для ВМ (з розрахунку максимум 32 vCPU на одне ядро процесора хоста) і видаліть жорсткий Floppy drive.

Тепер необхідно створити адміністратора Instant Clone. Додається з Active Directory користувач, він буде управляти комп'ютерами в контейнері (OU) Active Directory. Можна використовувати Computers (рисунок 3.3) або створити окремий для всіх клонів операційних систем окремий. Отже, користувач, який додається як адміністратор Instant Clone, має мати права на створення (Create Computer Objects) і видалення (Delete Computer Objects) комп'ютерів у цьому OU, а так само мати доступ для запису всіх властивостей (Write All Properties).

Призначимо користувача домену domain.com як адміністратора Instant Clone (рисунок 3.4) і переконаємося в тому, що ввімкнений View Storage Accelerator. Відкриваємо у вебконсолі Horizon [8–14]: *View*

Configuration –> Servers –> vCenter Servers, вибираємо vCenter Server, тиснемо кнопку Edit, а у вікні редагування налаштувань vCenter Server переходимо у вкладку Storage (рисунок 3.5). За бажанням розмір кешу за замовчуванням можна змінити, а так само задати індивідуально для кожного хоста (*Edit cache size –> Overrride default Cache size*).

Рисунок 3.4 – Створення Instant Clone адміністратора

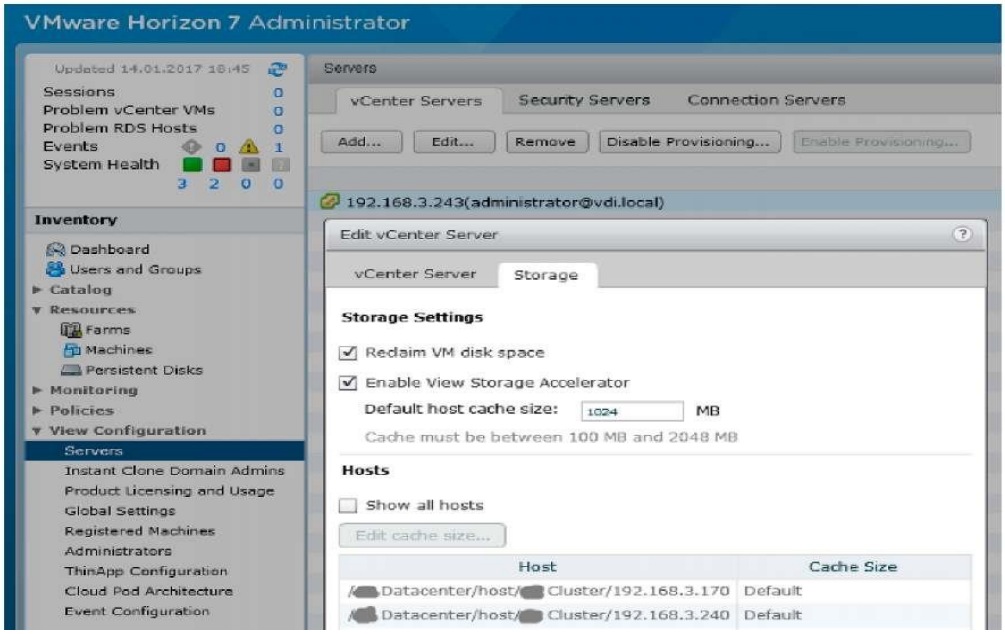


Рисунок 3.5 – Увімкнення View Storage Accelerator

Переходимо до створення пулу. Вибираємо Automated Desktop Pool і вибираємо змінне призначення (Floating), при якому ВМ зі створюваного пулу будуть призначатися випадково, кожен раз як користувач буде змінюватися. Створення Instant Clone Desktop Pool (рисунок 3.6). Обираємо Instant Clone, задаємо ID і коротке ім'я пулу (рисунок 3.7).

Рисунок 3.6 – Створення Instant Clone Desktop Pool

Рисунок 3.7 – Вибір Instant Clone, задавання ID і коротке ім'я пулу

Зверніть увагу, що налаштування 3D Render для Instant Clone будуть недоступні (рисунок 3.8). У вкладці Provisioning settings задаємо Naming pattern:

VIEW–IC– {n: fixed = 3}.

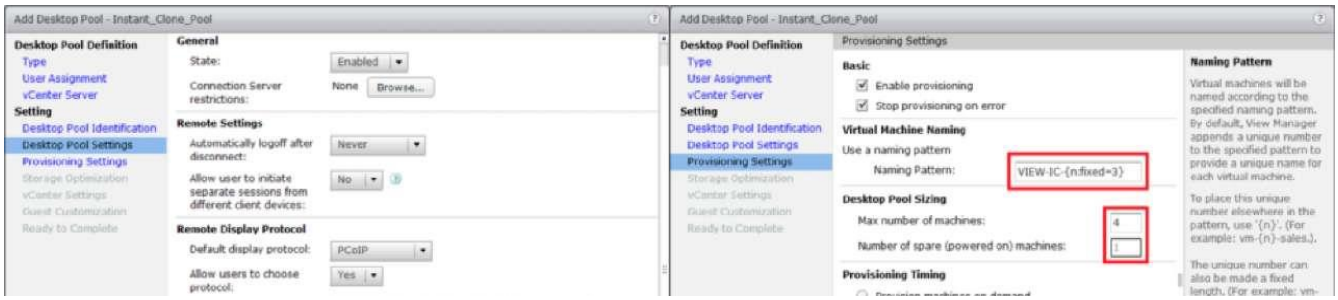


Рисунок 3.8 – Створення Instant Clone Desktop Pool

Отже, створені клони будуть називатися VIEW–IC–001, VIEW–IC–002 і т. д. Тут же вказуємо максимальну кількість машин (Max number of machines), а в Provisioning Timing – те, як вони будуть підготовлені:

– при виборі Provisioning machines on demand (розташувати машини на вимогу) ВМ будуть створені при спробі підключення користувача до пулу. Можна вказати мінімальну кількість машин, які будуть попередньо

підготовлені (Min number of machines) і будуть запасні – параметр Number of spare (powered on) machines;

– при виборі Provision machines up-front одразу буде створено максимальну кількість машин [8–14].

Для зберігання ВМ можна використовувати Virtual SAN. На відміну від повних клонів ВМ, тут надається можливіть розташувати реплікацію і ВМ пулу, до яких будуть підключатися користувачі (у нашому випадку це папка Instant_Clone_Pool), на різних datastore. Міттєві клони (Instant Clones) можна так само розміщувати на окремому хості і підключеному до нього datastore, але в цьому випадку ви втрачаєте підтримку vMotion, побудову відмовостійкості VMware High Availability (HA) і vSphere Distributed Resource Scheduler (DRS).

Вказуємо, як Parent VM підготовлений раніше Master Desktop Template. Створений знімок (snapshot) і розташування ВМ у вкладці vCenter Setting і зміни в порядку до Guest Customization. Тут вказуємо домен і згаданий раніше контейнер (OU), де будуть розташовуватися приєднані до Active Directory клони ВМ.

У полях Power-off скрипт можна вказати ім'я, шлях і параметри скрипта кастомізації системи, який буде запущений CloneP перед тим, як машину буде вимкнено. У полях post-synchronization аналогічно вказують параметри для запуску після створення ВМ або її синхронізації (поновлення образу) (рисунок 3.9) [8–14].

Рисунок 3.9 – Створення Instant Clone Desktop Pool

На фінальному екрані ставимо чек-бокс на опції Entitle users after this wizard finishes (або призначимо користувачів вручну пізніше), відкриваємо в vSphere Client: *Home –> Inventory –> VMs and Templates* або, переключившись у режим VMs and Templates, у вебконсолі управління vSphere побачимо ВМ, розташовані в папках (рисунок 3.10).

Рисунок 3.10 – Parent, Replica і Template VMs для Instant Clone

ClonePrepInternal TemplateFolder. cp–template–xxxx – шаблон віртуальної машини, який використовується для створення миттєвих клонів. Він був створений з вашого основного способу (Master Desktop Template) [8–14].

ClonePrepParentVmFolder. cp–parent–xxxx – батьківський образ. Кількість цих ВМ має відповідати кількості хостів у кластері. Кожен ESXi сервер буде мати одну таку ввімкнену ВМ для миттєвого клонування ВМ.

ClonePrepReplicaVmFolder. cp–replica–xxxx – реплікація, зразок ВМ для реплікації батьківського образу для кожного хоста ESXi.

ClonePrepResyncVmFolder. Якщо ви хочете оновити клони, то розмістіть тут новий образ для синхронізації. – Ітіапї_Clone_PooI.

VIEW_IC–NNN - відповідно самі клони віртуальних машин у пулі [8–14].

На рисунку 3.11 зображена схема процесу створення пулу миттєвих клонів з снапшотів Master Image до розподілу батьківського образу (Parent) по хостах ESXi з подальшим його клонуванням за лічені секунди. Після

створення батьківська ВМ розподіляє пам'ять вже завантаженої ВМ для створення кожного клону, тим самим забезпечуючи майже миттєву доступність ОС в уже завантаженому стані. Кожен з миттєвих клонів буде призначений для конкретного користувача з його логіном, а об’єм буде зростати пропорційно створеним змінам у цій ВМ відносно батьківського образу. Зміни в системі записуються в дельта-диск (а не батьківський) для кожного клону ВМ, тим самим забезпечуючи ізольованість і компактність дискових образів [8–14].

Після завершення процесу створення пулу у вкладці Inventory буде відображуватися статус Available для кожного клону пулу (рисунок 3.12), а в Horizon Client з'явиться доступний для підключення пул (рисунок 3.13).

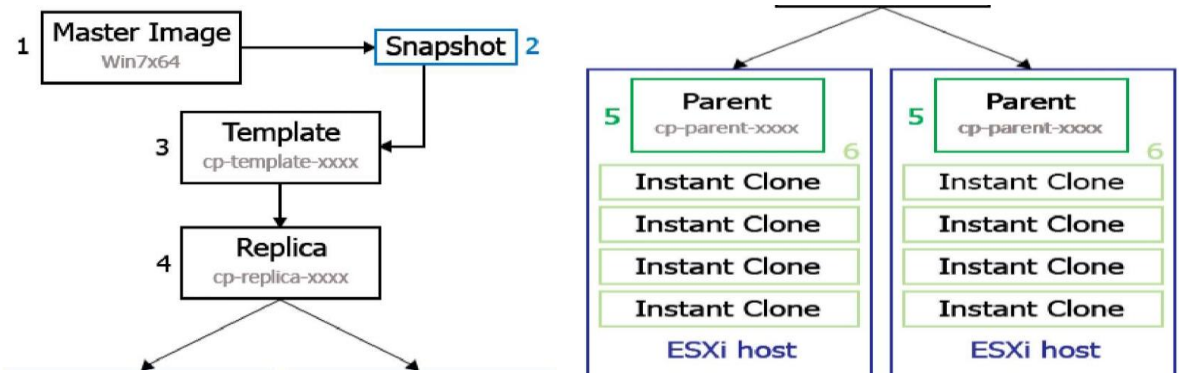


Рисунок 3.11 – Процес створення Instant Cloned у вигляді схеми

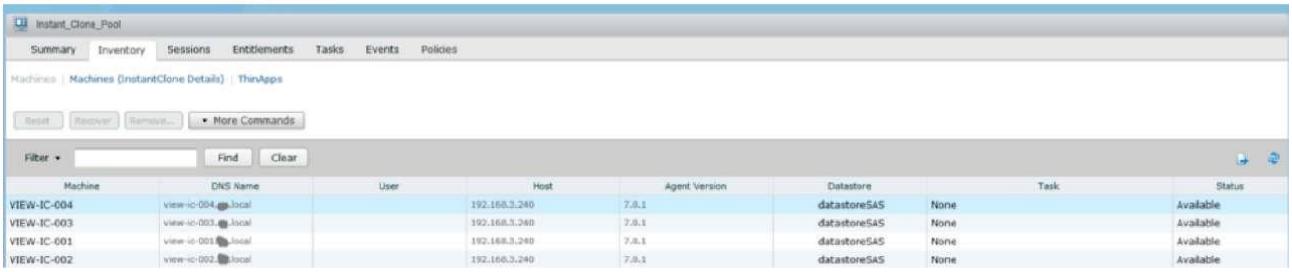


Рисунок 3.12 – Instant Clone Pool Inventory

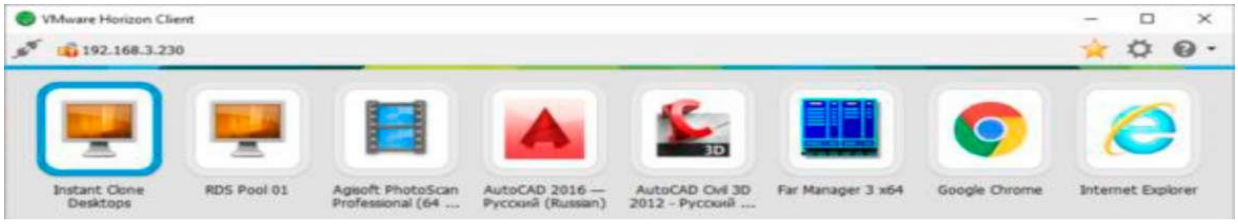


Рисунок 3.13 – VMware Horizon Client

Instant Clone Pool може масштабуватися за рахунок зміни кількості provisone клонів протягом досить короткого проміжку часу. Для того щоб оцінити можливості, збільшимо кількість клонів.

Відкриємо [8–14]: *Summary –> Edit –> Provisioning* і збільшимо кількість ВМ до 70 (рисунок 3.14). Процес створення клонів можна спостерігати в vSphere Client'e, перейшовши в режим огляду VMs and Templates і вибравши вкладку Virtual Machines (рисунок 3.15).

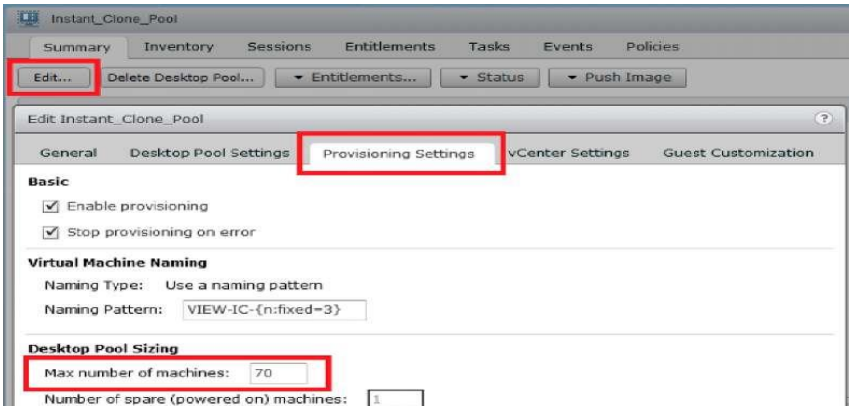


Рисунок 3.14 – Зміна кількості provisioning ВМ

Розширення пулу до 30 клонів зайняло приблизно 53 с, 50 клонів – 1 хв, а 70 – близько 2 хв. І хоча VMware офіційно заявляє, що для створення одного Instant Clone йде 1 с, у цьому випадку при оцінюванні враховувався відрізок часу до отримання цими ВМ IP-адреси за DHCP. Якщо не враховувати об’єм реплікації Master Template образу ОС (об'ємом 35 Гбайт), то 70 миттєвих клонів одразу ж після створення і кастомізації зайняли близько 250 Гбайт (при збільшенні активних сполук об’єм дещо

зросте). Пул розміщувався на одному двопроцесорному хості ESXi, а для зберігання використовувався локальний datastore – жорсткий диск SAS SSD. Мінімально рекомендований об’єм дискового простору для створення Instant Clone склав 420 Гбайб, а з урахуванням 50 % утилізації – 3010 Гбайт. Як приклад у таблиці на рисунку 3.16 наведено рекомендації для дискового простору для публікації і клонування ВМ об'ємом 60 Гбайт [8–14].

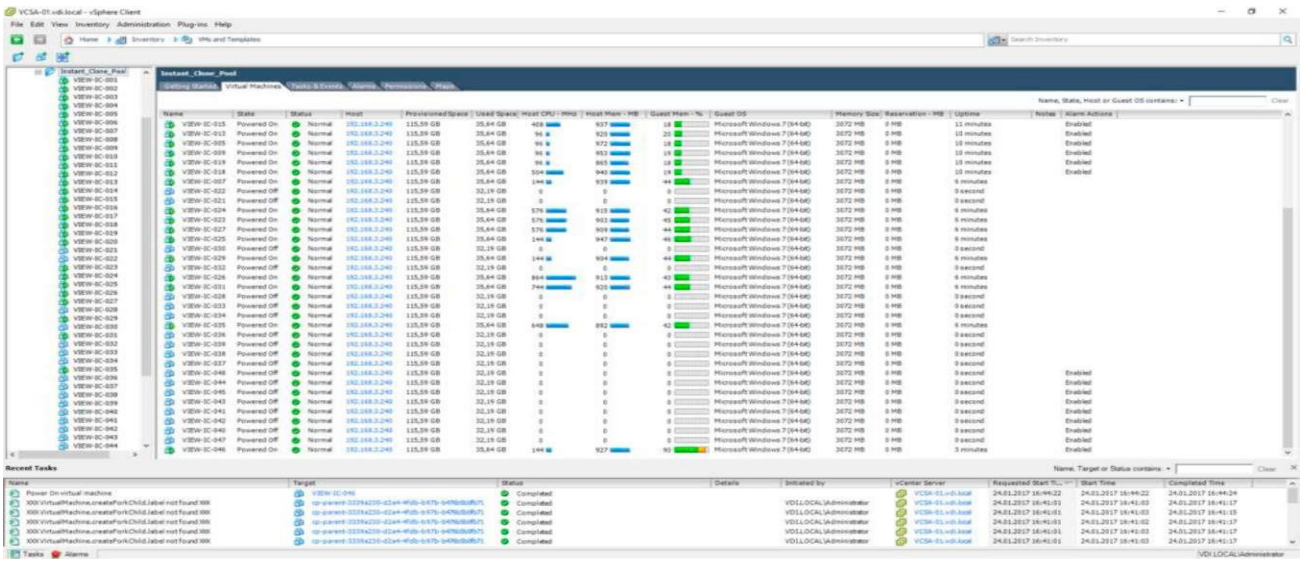


Рисунок 3.15 – Процес створення Instant Clone з 70 ВМ

NUMBER OF OBJECTS	VM TYPE	DISK SPACE USED
1	Master	60.0 GB
1	Internal template	55.0 MB
1 per datastore	Replica	21.0 GB
1 per ESXi host	Running parent	2.5 GB (primarily swap)
1 per desktop	Instant-clone desktop	2.5 GB (growth on use)

Рисунок 3.16 – Вимоги до дискового простору для 60 Гбайт ВМ

Крім об’єму і типу дискових накопичувачів, на швидкість створення клонів впливає кількість використовуваних хостів у кластері vSphere, використання Virtual SAN і, очевидно, пропускна спроможність ethernet, тому що процес реплікації шаблона ВМ відбувається по мережі.

3.2.3 Оновлення образу Instant Clone

Припустимо, нам необхідно змінити Master Template (встановити ПЗ або змінити якісь налаштування), а після цього відновити пул, не видаляючи його і по можливості не перериваючи процес роботи користувачів [8–14].

Наприклад, встановимо Google Chrome, Adobe Flash для Internet Explorer і якісь інші оновлення на Master Desktop Template (Win7x64). Після установлення ПЗ вимикаємо ВМ і створюємо snapshot (рисунок 3.17). Відкриваємо пул і переходимо у вкладку Summary, де вибираємо *Push Image –> Schedule* (рисунок 3.18).

У діалозі Schedule Push Image (рисунок 3.19) вибираємо snapshot, задаємо дату і час для публікації.

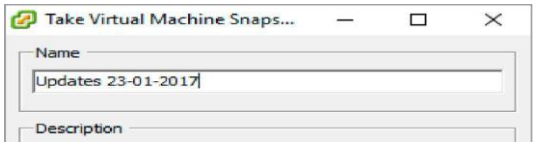


Рисунок 3.17 – Створення snapshot

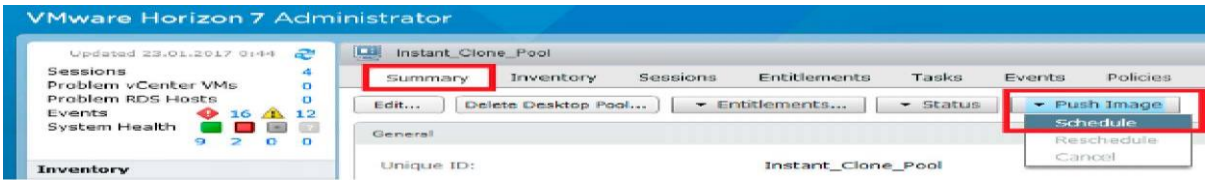


Рисунок 3.18 – Push Image

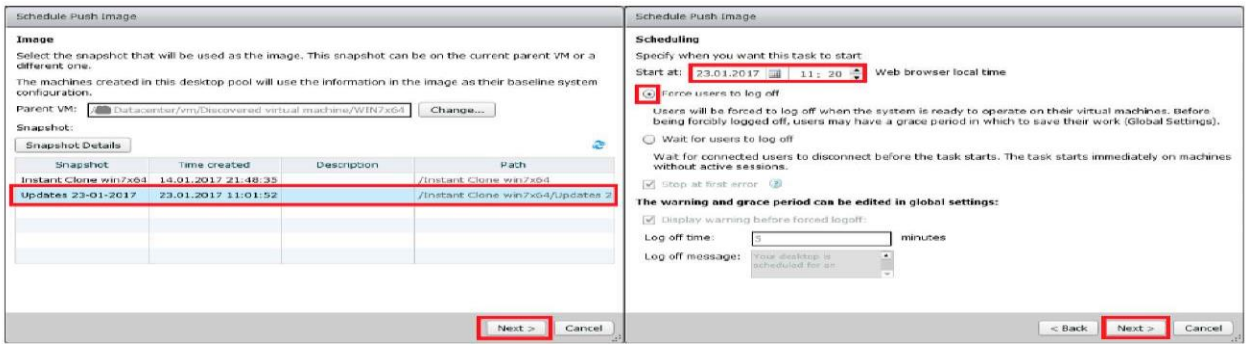


Рисунок 3.19 – Schedule Push Image

Далі вибираємо один з варіантів:

а) примусово завершити призначену для користувача сесію, тоді як налаштування тайм-ауту і текст повідомлення, який буде відображений для користувачів, задається в налаштуваннях Horizon View: *View Configuration –> Global Settings;*

б) чекати завершення сесії користувачів. Завдання буде автоматично запущено на віртуальних машинах без активних сесій користувачів [8–14].

Статус запланованого оновлення образу (Publishing (рисунок 3.20)) і його завершення (Published (рисунок 3.21)) будуть відображуватися у вкладці Summary.

Підключаємося під одним із заданих користувачів пулу до Instant Clone пулу на Elementary OS (установлення і налаштування Composer Server) і перевіряємо результат (рисунок 3.22).

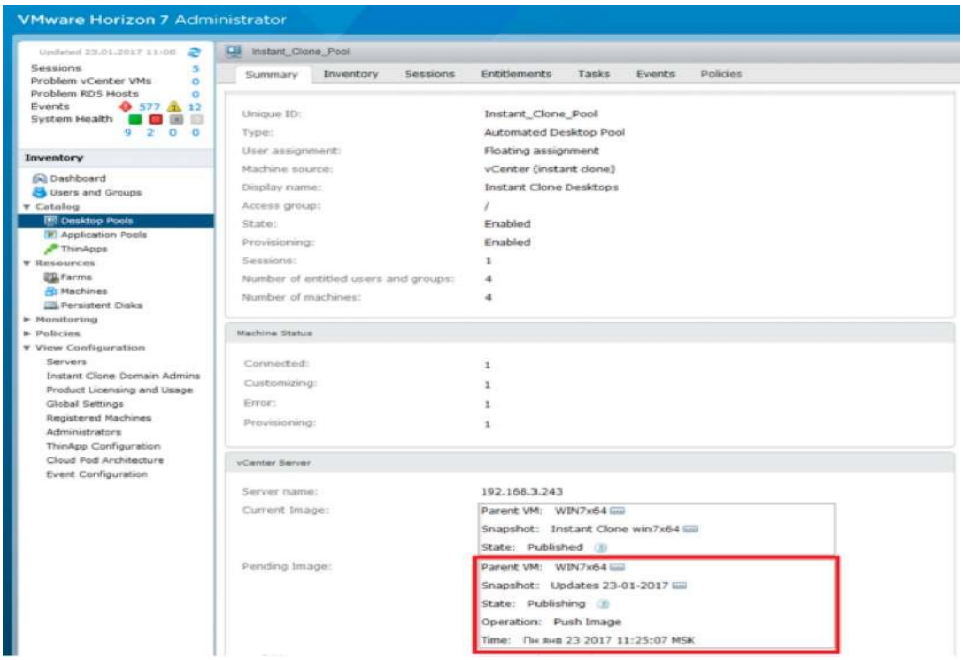


Рисунок 3.20 – Push Image Status

Рисунок 3.21 – Pushed Inaged Status

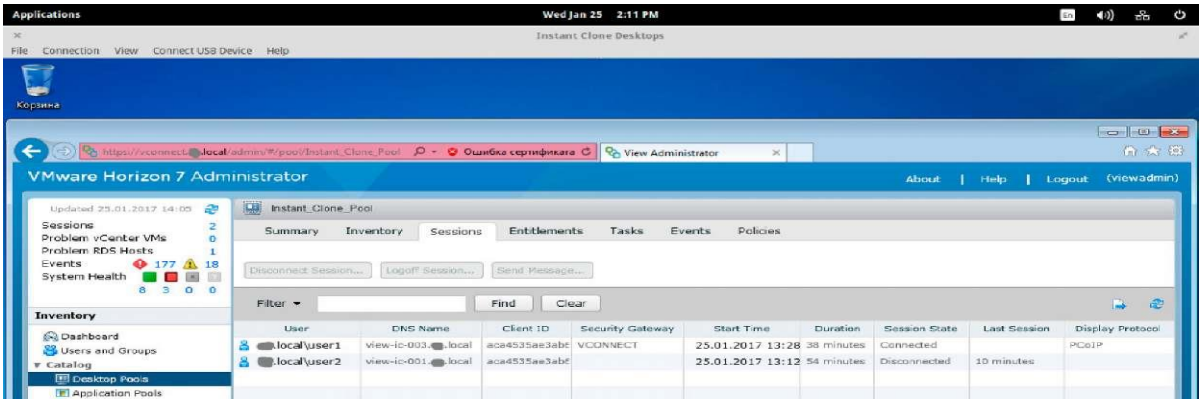


Рисунок 3.22 – Підключення до Instant Clone пулу на Elementary OS

Перш ніж почати використовувати Linked Clones, потрібно встановити Horizon Composer Server, який може бути встановлений як на один фізичний або віртуальний сервер разом з vCenter Server, так і окрему операційну систему. При цьому Composer Server не може бути встановлений на контролера домену Active Directory або разом з іншими компонентами Horizon View, такими як Connection Server або Security Server. Системні і апаратні вимоги для установлення Composer Server такі [8–14]:

– процесор: двоядерний Intel 64 / AMD 64 1.4 GHz (рекомендується чотириядерний з частотою не менше 2 GHz);

– оперативна пам'ять: не менше 4 Гбайт (для розгортання 50 і більше десктопів рекомендується 8 Гбайт);

– дисковий простір: 40 Гбайт (рекомендується 60 Гбайт);

– ОС: Windows Server 2008 R2 SP1 (Standart, Enterprise, або Datacenter) або Windows Server 2012 R2 (Standart або Datacenter) зі встановленим .NET Framework 3.5 SP1.

Підключення до СУБД. View Composer зберігає в базі даних з'єднання з vCenter Server і Active Directory, інформацію про розгорнуті Linked-clone десктопи і реплікації. Можна використовувати Microsoft SQL Server, встановлену разом з Composer Server і vCenter Server на одній операційній системі Windows Server, так і створити підключення по TCP / IP до окремо встановленої інстанції БД. Вимоги аналогічні Events Database [8–14].

3.2.4 Створення ODBC Database Connection для Composer Server

Як приклад розглянемо підключення Composer Server до вже встановленої на Connection Server (192.168.3.230, ілivconnect.domain.local) Microsoft SQL Express 2012 SP2 (рисунок 3.23).

Підключаємося в Microsoft SQL Management Studio та в першу чергу переконаємося, що перебуваємо в режимі змішаної автентифікації (рисунок 3.24). Створюємо БД ViewComposer, однойменний логін і зіставляємо з ним уже створену БД (рисунок 3.24).

Тепер переходимо до ВМ (або сервера), де буде встановлюватися View Composer.

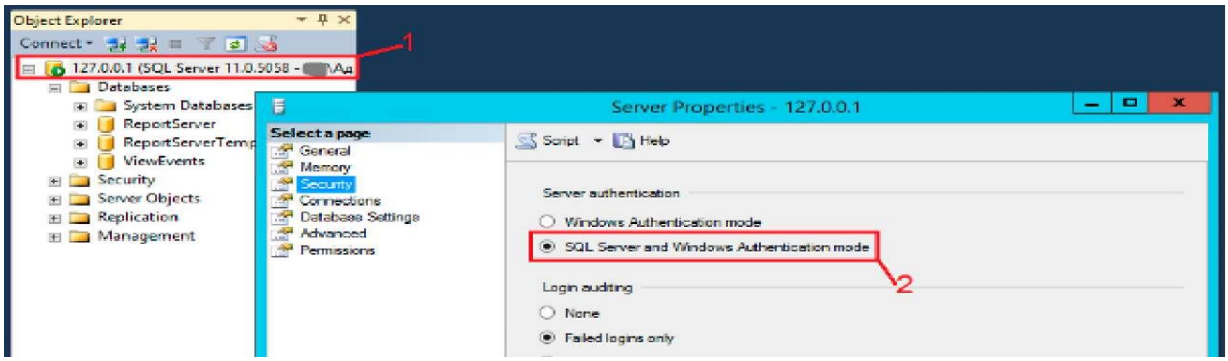


Рисунок 3.23 – SQL Server and Windows Authentication mode

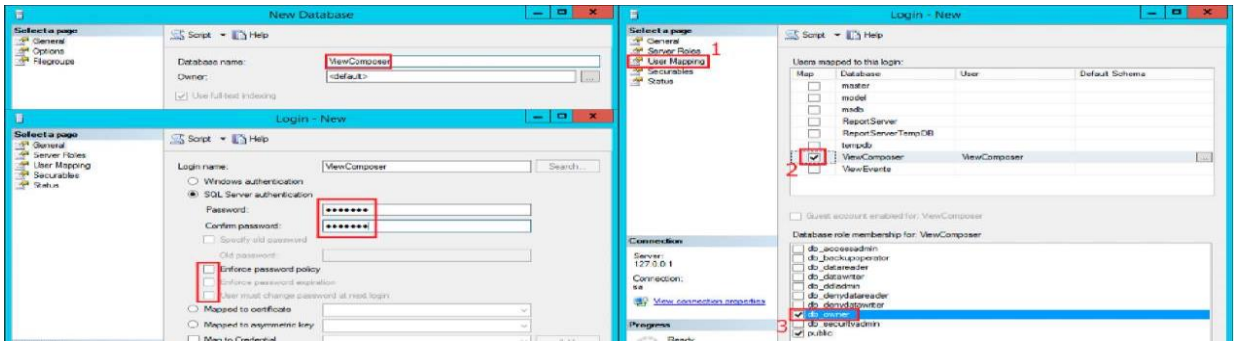


Рисунок 3.24 – Створення БД і логіна

Відкриваємо [8–14]: *Панель управління –> Адміністрування –> Джерела даних ODBC (64– розрядна) –> Системний DSN –> Додати –> SQL Server Native Client* і створюємо новий DSN: вводимо дані для

підключення до SQL Server (рисунок 3.25) і використовувану базу за замовчуванням. Створення DSN для SQL Server, установлення і додавання Composer Server (рисунок 3.26). Після завершення запускається тест для перевірки створеного джерела (рисунок 3.27). Викачуємо Horizon View Composer, запускаємо інсталятор і вказуємо створений вище DSN. Порт залишаємо за замовчуванням (рисунок 3.28).

При підключенні Composer Server ви можете побачити помилку:

A connection problem occurred between the Connection Server, View Composer and vCenter Server. Check that all the services are running and the ports and URL's are entered correctly. Можливі причини [8–14]:

– vCenter Single Sign-On Identity Source налаштований некоректно;

– vCenter Server зі неперетворюваним DNS або не підключений до контролера домену Active Directory.

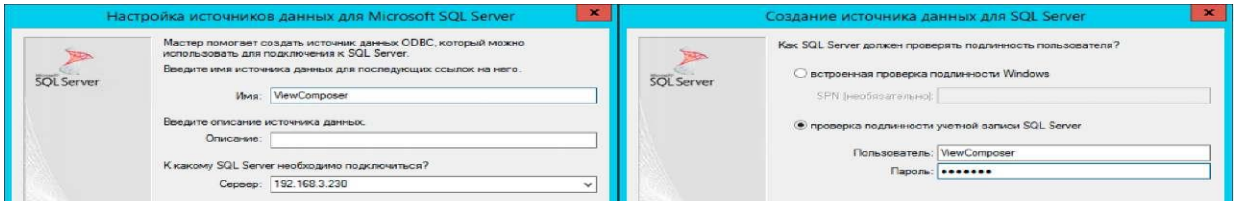


Рисунок 3.25 – Створення DSN для SQL Server

Рисунок 3.26 – Створення DSN для SQL Server

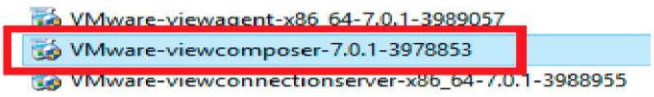


Рисунок 3.27 – Horizon 7 View Composer

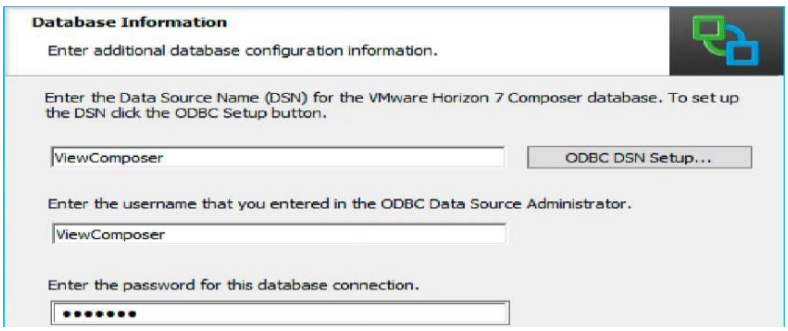


Рисунок 3.28 – Установлення Horizon View Composer

Для виправлення цієї помилки необхідно, щоб використовуваний для приєднання Composer Server обліковий запис домена domain.local володів правами адміністратора цього домену domain.local і був призначений адміністратором True SSO. Так само необхідно, щоб vCenter Server був підключений до домен-контролера domain.local [8–14].

Створимо обліковий запис domain.local\domain_admin з правами адміністратора цього домену та порівнюємо права адміністратора True SSO (vdi.local) (рисунки 3.29, 3.30).

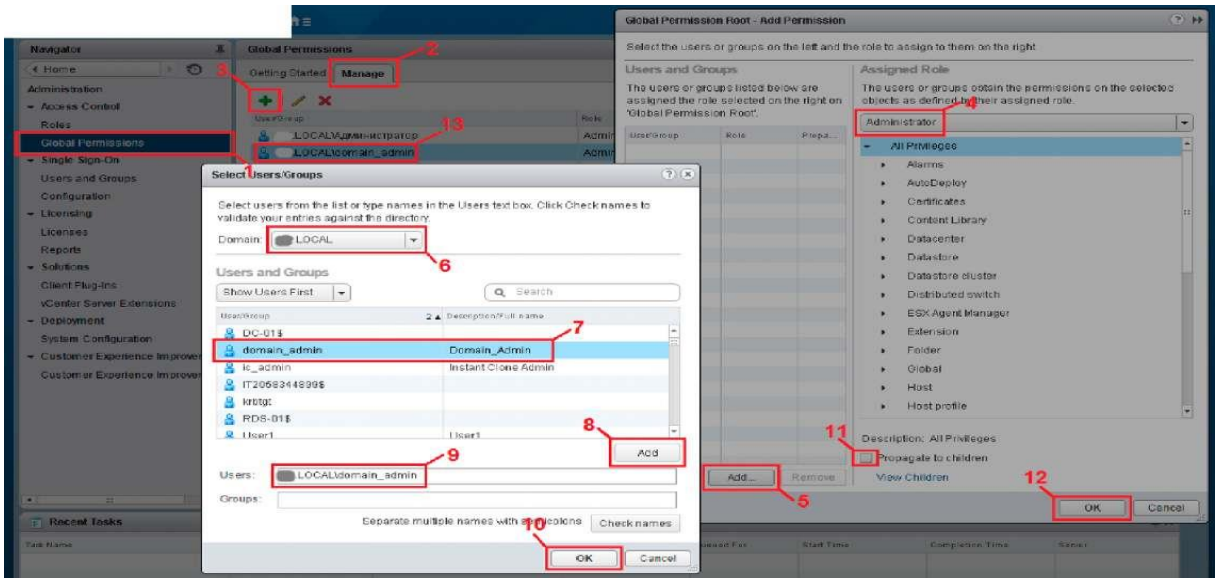


Рисунок 3.29 – Зіставлення і призначення прав для облікового запису

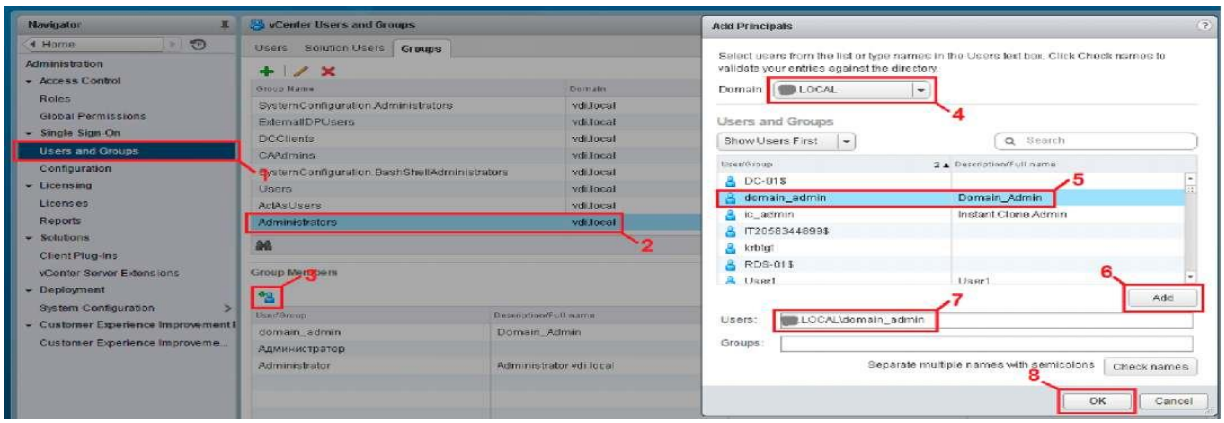


Рисунок 3.30 – Зіставлення і призначення прав для облікового запису

Тепер під цим обліковим записом підключаємо Composer Server до Connection Servery, відкриваємо: View Configuration –> Server –> vCenter Servers –> Edit. У цьому прикладі Composer Server встановлювався окремо від vCenter Server, тому вибираємо варіант Standalone View Composer Server, вводимо IP- адресу, логін і пароль для створеного користувача domain_admin домену domain.local (рисунок 3.31) і ігноруємо помилку, пов'язану з сертифікатом. Зверніть увагу, що для роботи Composer Server необхідно vCenter Server підключити з тим самим обліковим записом, що і Composer Server [8–14].

Інформацію про підключений Composer Server можна подивитися при повторному відкритті опції Edit (рисунок 3.31) або на Dashboard (рисунок 3.32).

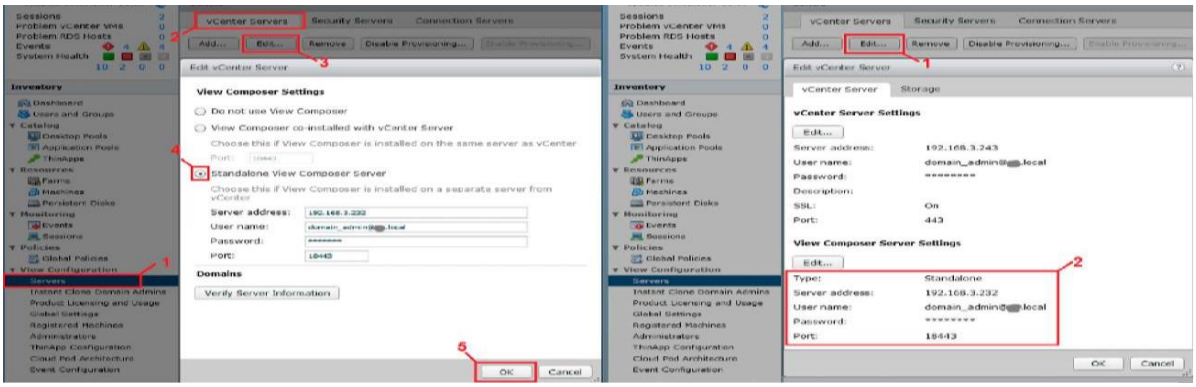


Рисунок 3.31 – Підключення Composer Server

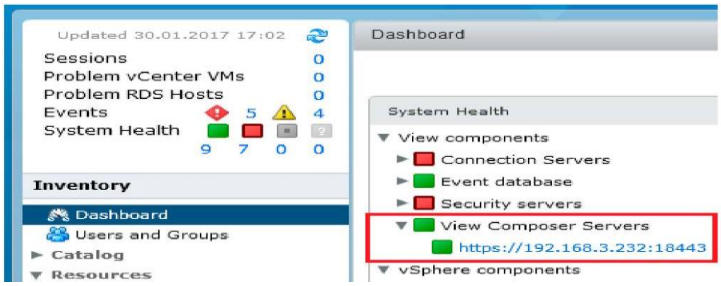


Рисунок 3.32 – View Composer–Dashboard

Для доступу до домену, у якому будуть створюватися ВМ Linked Clone Desktop Pool, необхідно додати цей домен (у цьому прикладі – domain.local). В іншому випадку майстер створення пулу десктопів при спробі створити Linked Clone Desktop Pool відобразить порожній список. Відкриваємо пункт Edit в опції View Composer Server Settings (рисунок 3.33), вибираємо: *Verify Server Information –> Add*. Вказуємо домен, логін доменного адміністратора (domain_admin) і пароль.

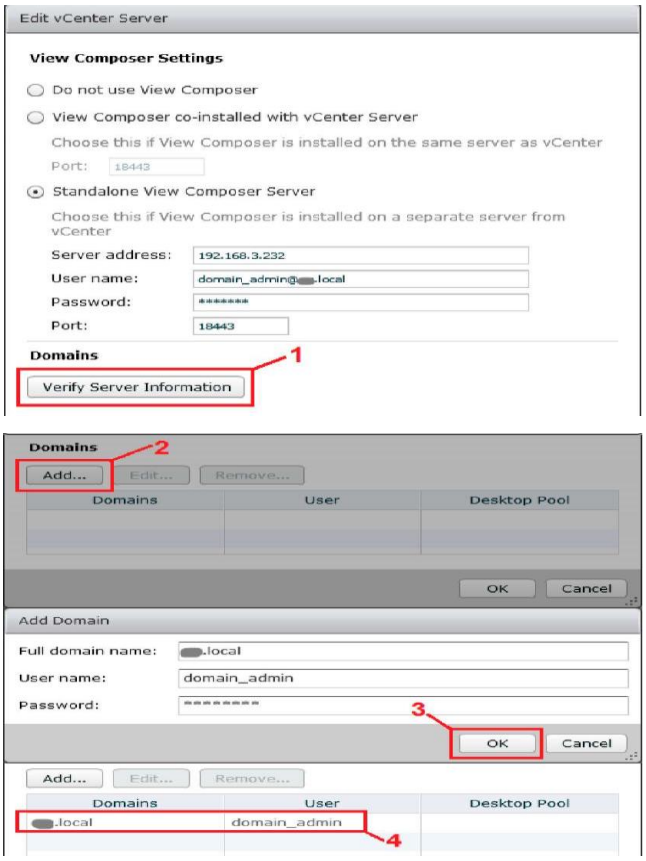


Рисунок 3.33 – Додавання домену для Linked Clones

3.2.5 Створення Linked-Clone Desktop Pool

Для створення Linked-Clone пулу можна використовувати той самий Master Desktop Template з Windows 7 [8–14]. Головне, щоб при установленні View Agent був обраний VMware Horizon View Composer Agent (рисунок 3.34). Якщо ви плануєте використовувати перенаправлення файлу підкачки і тимчасових файлів (Disposable File Redirection), то перед створенням snapshot видаліть з системи pagefile.sys для запобігання його дублюванню на всіх створюваних клонах. Тепер, коли Master Desktop Template (або Parent VM) підготовлений, вимикаємо ВМ, створюємо snapshot і переходимо до створення пулу: відкриваємо у вебконсолі Horizon: *Catalog –> Desktop Pools –> Add.*

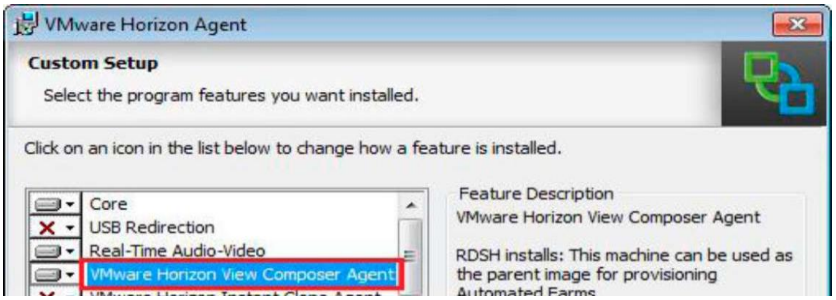


Рисунок 3.34 – Установлення Horizon View Composer Agent

На відміну від миттєвих клонів, Linked-Clone дають змогу налаштувати для створюваних десктопів графічну акселерацію, а так само використовувати NVIDIA GRID у режимі vSGA. При створенні Linked-Clone Desktop Pool можна вибрати як змінне призначення користувачів (Floating), так і виділене (Dedicated). Floating і Dedicated Assignment не однакові за своїми можливостями:

– вибір Floating Assignment дає змогу користувачам ініціювати окремі сесії для кожного з клієнтських пристроїв (Allow user to initiate separate sessions from different client devices);

– вибір Floating Assignment дає змогу видаляти або оновлювати (refresh) ВМ після завершення користувальницької сесії (log-off);

– вибір Dedicated Assignment дає змогу оновлювати диск операційної системи після закінчення сесії (Delete or refresh machine on logoff) [8–14].

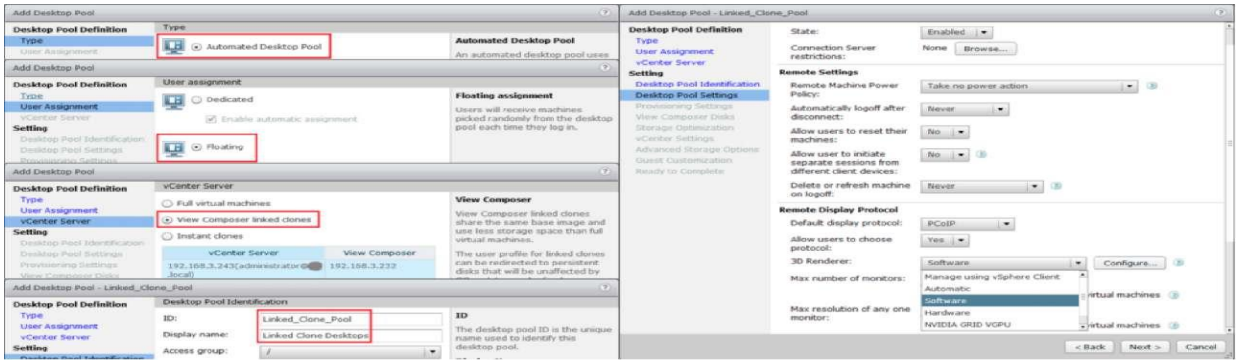
Ознайомитися з можливостями Linked-Clone Desktop Pool, а також різними режимами User Assignment можна в таблиці на рисунку 3.35.

Setting	Linked-Clone Pool, Dedicated Assignment	Linked-Clone Pool, Floating Assignment
State	Yes	Yes
Connection Server restrictions	Yes	Yes
Remote machine power policy	Yes	Yes
Automatically logoff after disconnect	Yes	Yes
Allow users to reset their machines	Yes	Yes
Allow user to initiate separate sessions from different client devices		Yes
Delete or refresh machine on logoff		Yes
Refresh OS disk after logoff	Yes	
Default display protocol	Yes	Yes
Allow users to choose protocol	Yes	Yes
3D Renderer	Yes	Yes
Max number of monitors	Yes	Yes
Max resolution of any one monitor	Yes	Yes
Adobe Flash quality	Yes	Yes
Adobe Flash throttling	Yes	Yes
Override global Mirage settings	Yes	Yes
Mirage Server configuration	Yes	Yes

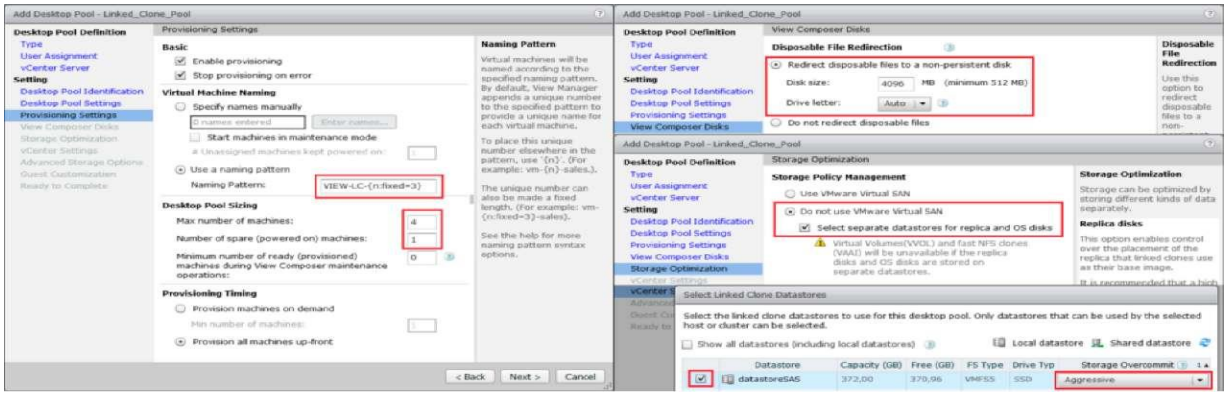
Рисунок 3.35 – Порівняння режимів User Assignment для Linked-Clone Desktop Pool

Опція Disposable File Redirection дає змогу перенаправляти на окремий віртуальний диск файл підкачки (page file) і тимчасові файли (рисунок 3.36).

Для кожної ВМ пулу створюється окремий vmdk (рисунок 3.37), який буде замінений оригінальними файлами батьківської ВМ (parent VM), як тільки ВМ буде вимкнена (power off). Можна так само задати букву і розмір диска, який має бути більше, ніж розмір файлу підкачки ВМ батька. Використання цієї опції дає змогу сповільнити збільшення займаного об’єму клону ВМ. Зверніть увагу, що не слід вибирати букву диска, вже використану в батьківській ВМ [8–14].



а)



б)

Рисунок 3.36 – Створення Linked-Clone Desktop Pool

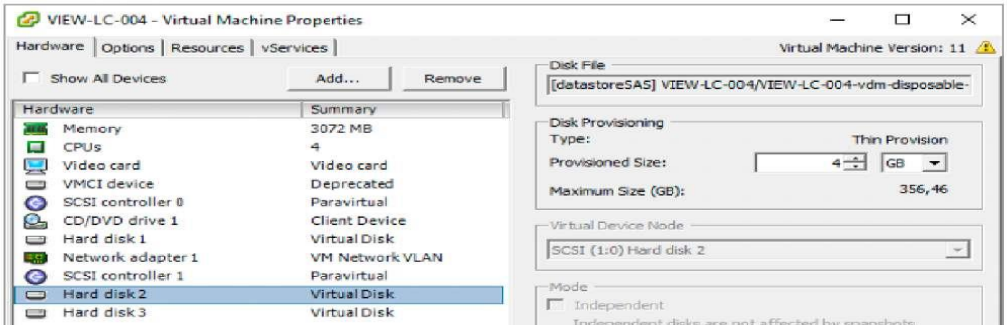


Рисунок 3.37 – Hard disk 2-Disposable File Redirection vmdk

Налаштування Provisioning settings, View Composer Disks і Storage Optimization аналогічні, як і для миттєвих клонів. На вкладці vCenter Server можна задати datastore для зберігання клонів і реплікованої ВМ (якщо поставлений чек-бокс на Select separate datastore for replica and OS disks).

Для кожного datastore можна задати свій рівень Storage Overcommit, налаштування, що дає змогу зберігати клони, сумарний об’єм яких більше, ніж ліміт datastore. Ця опція доступна тільки для Linked-Clone і дає змогу задати, у скільки разів може перевищувати об’єм створюваних клонів об’єм datastore, якби кожен зі створюваних клонів був повним клоном віртуальної машини [8–14]:

– None – Storage Overcommit вимкнений;

– Conservative – у чотири рази перевищує об’єм datastore. Це значення за замовчуванням;

– Moderate – у сім разів перевищує об’єм datastore;

– Aggressive – у пятнадцять разів перевищує об’єм datastore;

– Unbounded – View не обмежує кількість ВМ, що розміщуються на datastore.

Вибирайте це значення, тільки коли сховище має достатньо місця для майбутнього зростання об’єму, займаного ВМ.

Фактично Storage Overcommit впливає на інтенсивність використання сховища, а калькуляція необхідного об’єму для кожного з параметрів відображується при виборі datastore і параметра Storage Overcommit в інтерактивному режимі (рисунок 3.38).

Налаштування Storage Accelerator аналогічні, як і при створенні миттєвих клонів. На вкладці Guest Customization вибираємо домен і контейнер для розташування створюваних клонів. Можна так само задати параметри додаткової кастомізації клону (QuickPrep, або SysPrep) (рисунок 3.39). По завершенні процесу створення Linked-Clone Pool в Horizon Client буде доступне нове підключення (рисунок 3.40), а статус ВМ відображуватиметься у вебконсолі Horizon у вкладці Inventory. Для Linked-Clones пулів є кілька специфічних функцій: Refresh, Recompose і Rebalance (рисунок 3.41);

Рисунок 3.38 – Калькуляція необхідного вільного дискового простору

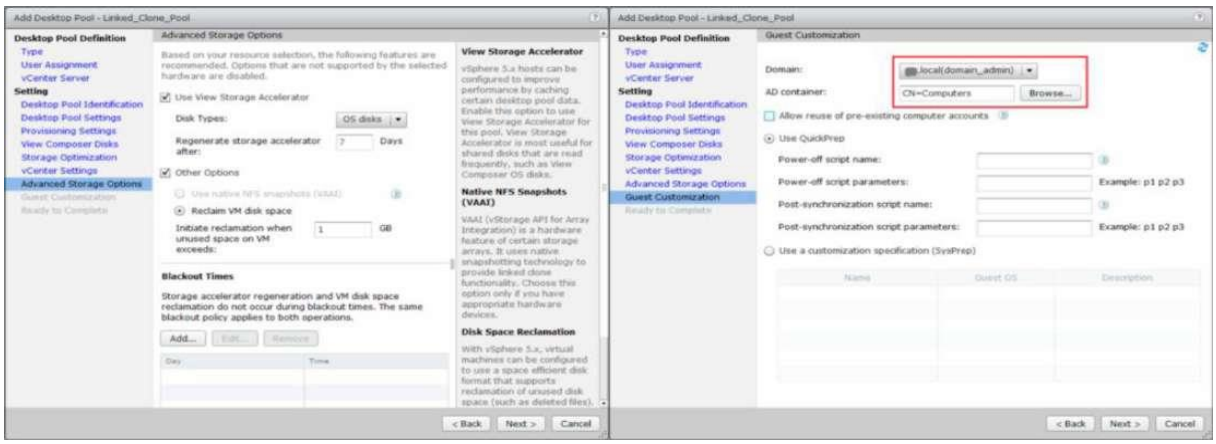


Рисунок 3.39 – Створення Linked–Clone Desktop Pool RDS Desktop Pool

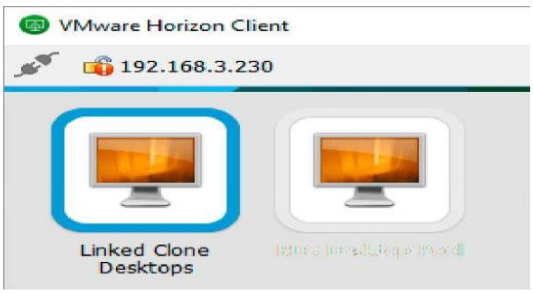


Рисунок 3.40 – Linked Clone Desktops

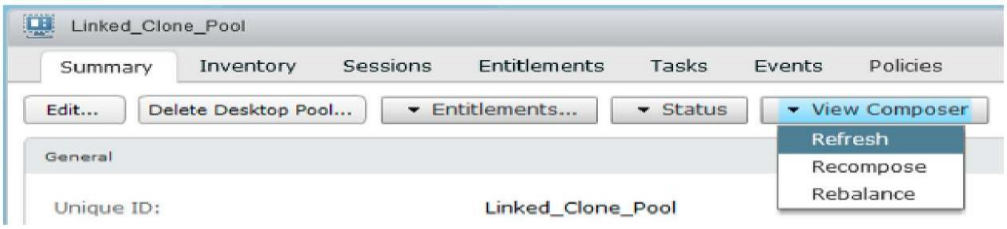


Рисунок 3.41 – Refresh, Recompose і Rebalance

– Refresh дає можливість скинути дельта-диски в початковий стан. Процес дає змогу повернути ВМ у початковий стан, коли був створений snapshot, і будь-які зміни користувача будуть скинуті. Ця функція може бути так само ввімкнена налаштуваннями пулу. Для цього у вебконсолі Horizon необхідно перейти: *Catalog –> Desktop Pools –> Summary –> Edit –> Desktop Pool –> Delete or refresh machine on logoff* і виставити в значення Refresh Immediately. Операція Refresh разом з вибором варіанта Force users to log off має

примусовий характер, оскільки користувачі, підключені до декстопів, будуть отримувати повідомлення, що змушує їх розлогінитися і вийти з робочого оточення, після чого ВМ буде відключена і створені нові дельта-диски. Як варіант, можна використовувати Wait for users to log off, у цьому випадку оновлення ВМ почнеться одразу ж після завершення користувальницької сесії;

– Recompose дає можливість створити новий replica-образ і замінити його на всіх ВМ пулу, у результаті чого всі зміни дельта-дисків будуть втрачені, і користувачі отримають оновлений образ (який, наприклад, містить нові додатки або поновлення системи). Операція Recompose разом з вибором варіанта Force usere to log off так само має примусовий характер, оскільки користувачі активних сесій будуть отримувати повідомлення про завершення сесії, тоді як при наступному логіні вони отримають новий десктоп;

– Rebalance використовується для перерозподілу десктопів за кількома datastore, якщо в якийсь момент виявилося, що ваші десктопи займають більший об'єм на якомусь одному datastore, ніж на іншому. Rebalance є виключно storage management командою і дає змогу рівномірно розподілити десктопи серед сховищ (datastores).

Зверніть увагу, що для забезпечення безперервної роботи пулу на випадок Maintance-операцій можна задати запасну кількість ВМ: параметр Minimum number of ready (provisioned) machines during View Composer maintenance operations у властивостях пулу (вкладка Provisioning settings).

Що стосується порівняльних характеристик або свого роду benchmark, то розширення Linked-Clone Desktop пулу з 4 Provisioned ВМ до 30 зайняло 88 с, а зі 4 до 50 ВМ – 3 хв. При цьому 30 клонів з неактивними сесіями зайняли 121 Гбайт дискового простору на datastore, а 50–249,9 Гбайт, тоді як розмір swap-file для Parent VM становив 3072 Мбайт, а реплікований образ зайняв додатково 35 Гбайт [8–14].

3.2.6 Порівняння Ins tant Clones і Linked-Clones

Створення Instant Clone Pool набагато швидше Linked-Clone Pool, займає трохи менше процесорних ресурсів і на порядок менше дискових операцій введення / виведення. Етапи клонування ВМ при використанні

обох технологій можна подати графічно у вигляді схеми (рисунок 3.42), тоді як сам процес створення пулу можна умовно поділити на дві частини [8–14]:

– Publishing (або Priming) – процес створення Master Image, що займає від 7 до 40 хв залежно від об'єму Master Image;

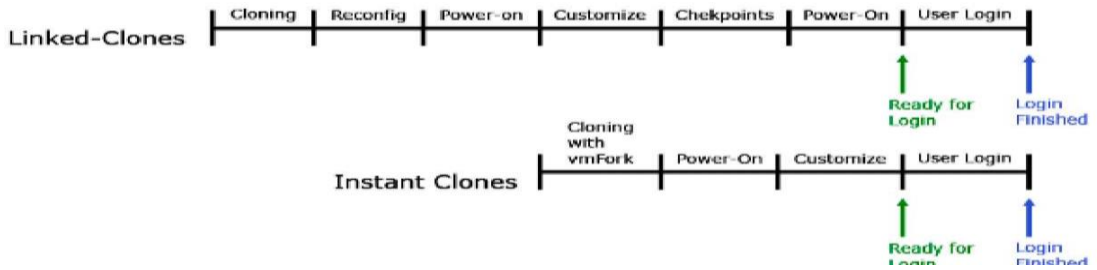


Рисунок 3.42 – Linked-Clones vs. Instant Clones

– Provisioning – створення клонів пулу десктопів, що займає від 1 до 2 с на один десктоп для Instant Clone і від 2 до 10 с для Linked Clones. Чим більше кількість ВМ у пулі, тим більша різниця в часі, яку займає при створенні Instant Clones і Linked-Clones. Фактично всі процеси при створенні пулу можна умовно поділити на три категорії звернення до vCenter (vCenter calls).

Клонування (Cloning):

– увімкнення і вимкення ВМ (Power cycle);

– реконфігурація (Reconfiguration).

І хоча vmFork має вище пікове навантаження на vCenter, ніж View Composer, саме навантаження більш короткочасне за рахунок зменшення кількості звернення до vCenter (рисунок 3.43). Instant Clones не потребують оновлення (Refresh, або Recompose) і балансування (Rebalance). Коли користувач завершує сесію (log out), робочий стіл буде видалений і відновлений з останнього snapshots. Краще використовувати Instant Clone пул там, де це можливо [8–14].

Тип вызова vCenter (vCenter Calls)	Linked-Clones	Instant Clones
Cloning	1 clone call	1 vmFork call
Power cycle	2 power cycle calls	-
Reconfiguration	3-4 reconfiguration calls	-
Общая нагрузка на vCenter	Выше	Ниже

Рисунок 3.43 – Типи виклику vCenter при створенні Linked і Instant Clones

Instant Clone десктопи підтримують:

– тільки одного користувача (Floating Pool) змінний пул (Floating Pool) – призначені пули (Dedicated pools) не підтримуються;

– максимум 1 vCenter Server;

– підтримується тільки один VLAN;

– підтримуються лише десктоп ОС: Windows 10 (32/64 bit) і Windows 7 SP1 (32/64 bit);

– максимум два монітори з роздільною здатністю до 2560 x 1600;

– як сховища підтримуються VMFS і Virtual SAN (VSAN) сторейджи;

– до 2000 Instant Clone десктопів на один пул;

– підтримується vMotion, vSphere DRS і vSphere HA.

Instant Clone десктопи не підтримують:

– Persona Management;

– RDSH;

– 3D Graphics (NVIDIA GRID): тільки обмежений режим vSGA замість vGPU;

– Virtual Volumes, VAAI і NFS сторейджи;

– Disposable disk;

– Persistent disks, якщо, наприклад, ви хотіли б використовувати App Volumes і User Environment Manager (UEM);

– PowerCLI;

– Storage vMotion для Instant Clone [8–14].

3.2.7 Створення SSL-сертифікатів

Переключимо з'єднання Connection і Security Server'a з дефолтного на використання довіреної SSL сертифіката. У тестовому середовищі створення сертифікатів можна пропустити, в інших випадках для установлення безпечного з'єднання необхідно створити сертифікати для Connection і Security Server. Теоретично для створення сертифікатів можна використовувати різні онлайн-сервіси, такі як, наприклад, StartSSL, службу сертифікації Active Directory у Windows або створити self-singed сертифікат за допомогою OpenSSL. У цьому прикладі будемо використовувати службу сертифікації Active Directory (Microsoft Certificate Authority) [8–14].

3.2.8 Установлення Microsoft Certificate Authority (центр сертифікації)

У першу чергу потрібно створити користувача, від імені якого працюватимуть центр сертифікації Active Directory (ЦС): переходимо в оснастку AD – користувачі й комп'ютери і створюємо користувача CertAdmin з правами адміністратора домену. Далі переходимо до установлення компонентів, де встановлюємо Службу сертифікатів Active Directory (Active Directory Certificate Services) (рисунок 3.44). Служба реєстрації в центрі сертифікації через Інтернет (Certificate Authority Web Enrolment) нам не знадобиться, якщо тільки ви не збираєтеся виписувати сертифікати через вебсервер IIS. Після завершення установлення необхідно налаштувати служби: натискаємо в повідомлення (рисунок 3.45) і переходимо до вказівки установлень у Майстрі установлень – задаємо облікові дані (Domain\CertAdmin), ставимо чек-бокс на центрі сертифікації і вибираємо [8–14]: *ЦС підприємства –> Кореневий ЦС –> Створити новий закритий ключ –> RSA # MSKSP 2048 SHA1*.

Про всяк випадок даємо змогу взаємодіяти з адміністратором і вказуємо період дії сертифіката, наприклад 10 років (рисунок 3.46).

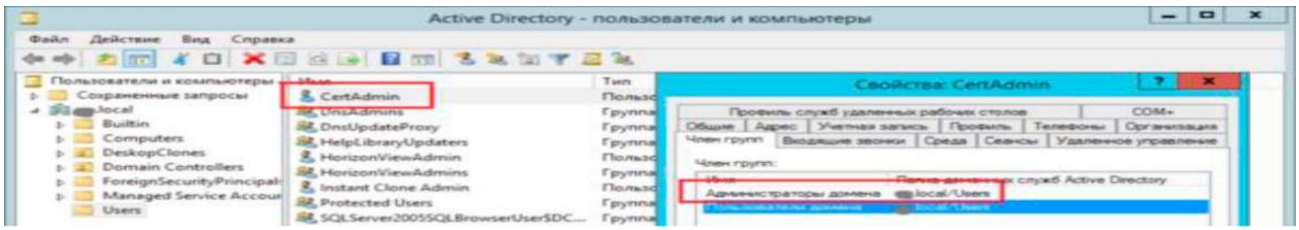


Рисунок 3.44 – Створення користувача CertAdmin

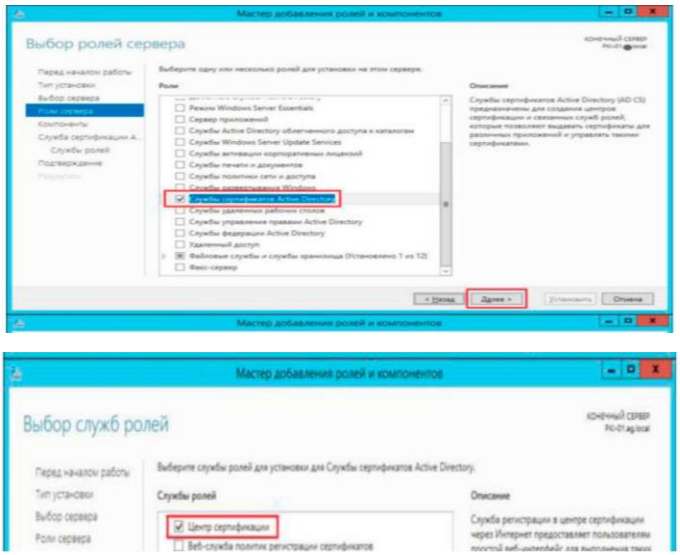


Рисунок 3.45 – Установлення служб центру сертифікації

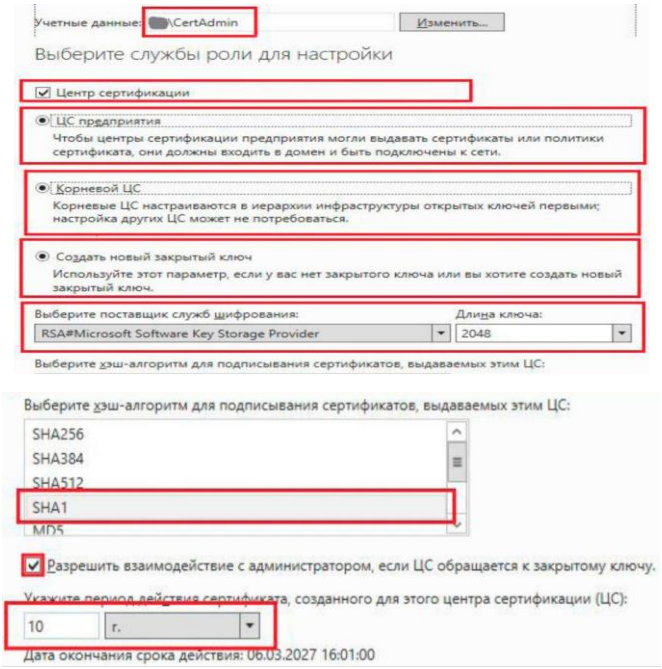


Рисунок 3.46 – Майстер налаштування служб центру сертифікації

Після завершення налаштування ЦС набираємо в командному рядку mmc і у відкритій консолі набираємо: *Файл –> Додавання і видалення оснасток –> PKI підприємства –> Додати –>Ok;* в оснастці, що з'явилася, заходимо в параметри правою кнопкою і вказуємо параметри CRL (рисунок 3.47). Зміна цих параметрів вимкне ВМ з розгорнутим PKI, при цьому видані сертифікати не виявляться простроченими [8–14].

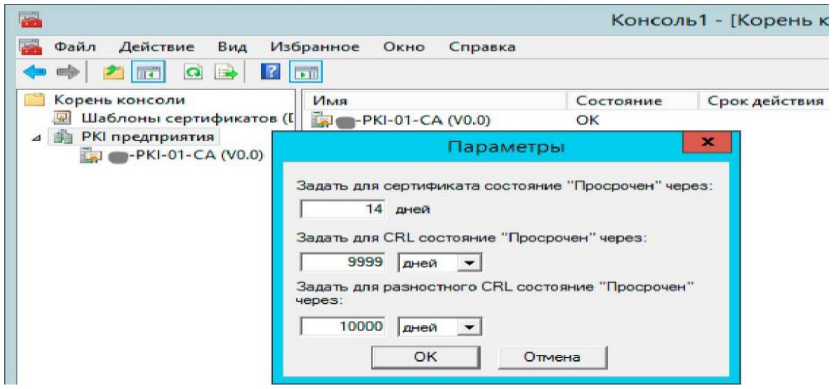


Рисунок 3.47 – Задавання параметрів CRL

Запит сертифікатів через консоль mmc. Запитаємо сертифікат для View Connection Server через оснастку (mmc). Підключаємося до Connection Server і вводимо в командному рядку mmc.exe та додаємо оснастку: *Файл –> Додати або видалити оснастку (Cntrl + M) –> Сертифікати –> облікового запису комп 'ютера –> Готово – локальним комп'ютером –> Готово – > Ок* або можна відразу запустити потрібну оснащення командою certlm.msc. Відкриваємо властивості сертифіката за замовчуванням (або раніше встановлених сертифікатів) і прибираємо всі, що містяться в полі Ясна, ім'я (рисунок 3.48).

Зверніть увагу, що можливий тільки один сертифікат із зрозумілим ім'ям vdm, який зараз і створимо за допомогою Служби сертифікації Active Directory, відкриваємо (рисунок 3.49): *Приватне –> Все завдання –> Просити новий сертифікат...* У відкритому майстрі запитуємо сертифікат для комп'ютера (рисунок 3.50) і відкриваємо властивості запитуваного сертифіката Security Center [8–14].

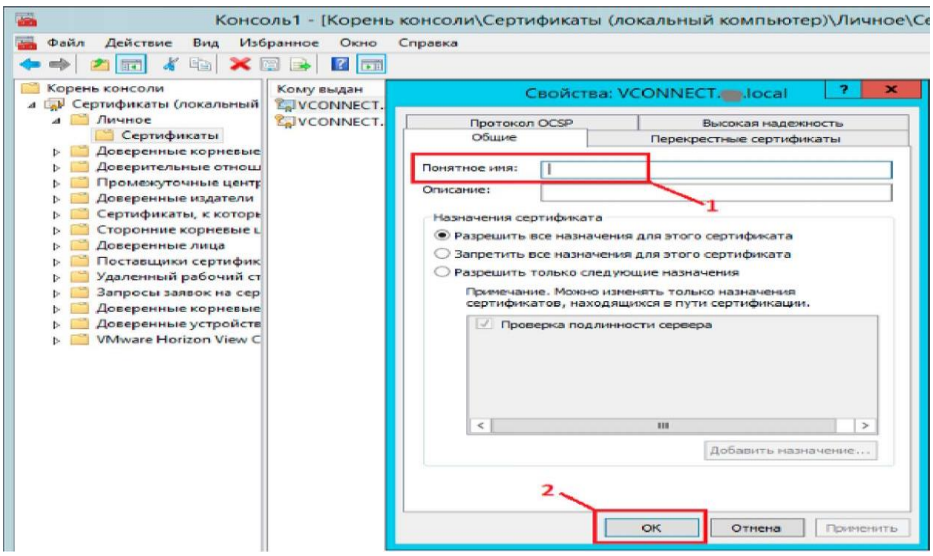


Рисунок 3.48 – Редагування імені сертифікатів за замовчуванням

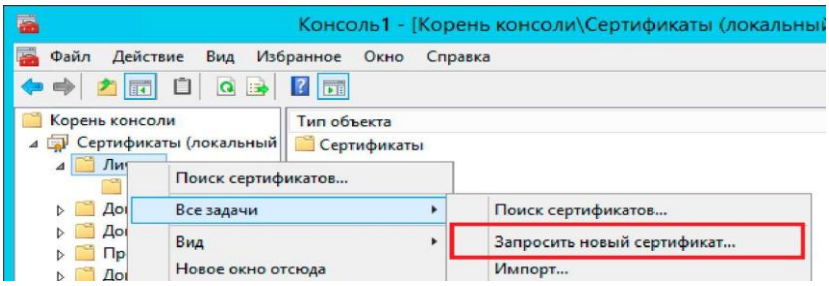


Рисунок 3.49 – Запит нового сертифіката

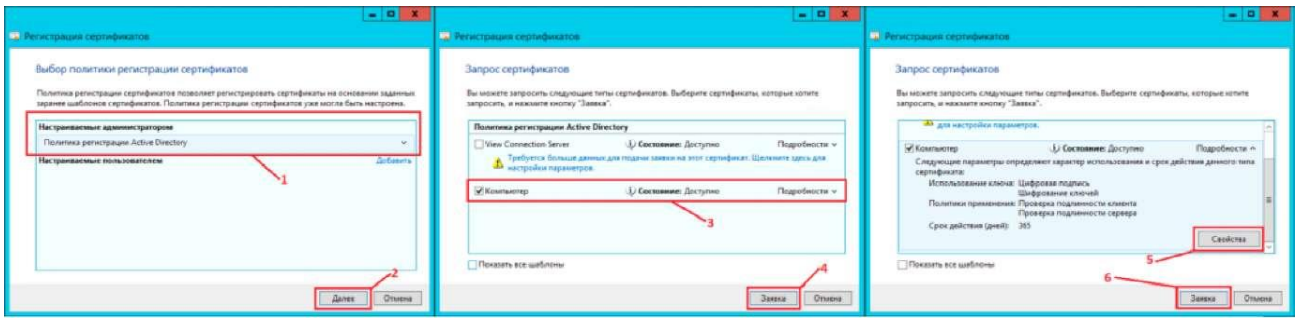


Рисунок 3.50 – Реєстрація сертифіката для Connection Server

Вказуємо як Ясне ім'я – vdm, встановлюємо розмір ключа рівним 2048, що дає змогу експортувати закритий ключ (рисунок 3.51).

Зверніть увагу, що для працездатності сертифіката в його властивостях має бути обов'язково дозволений експорт. Можна перевірити влас-

тивості запитаного сертифіката: вибираємо створений сертифікат з ім'ям DOMAIN.PKI_NETBIOS–CA і пробуємо експортувати (рисунки 3.52, 3.53).

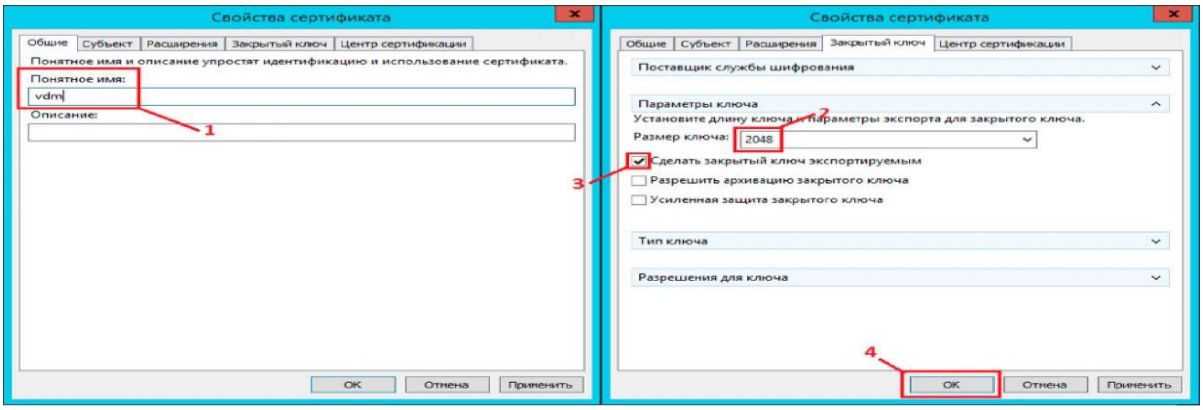


Рисунок 3.51 – Властивості реєстрованого сертифіката

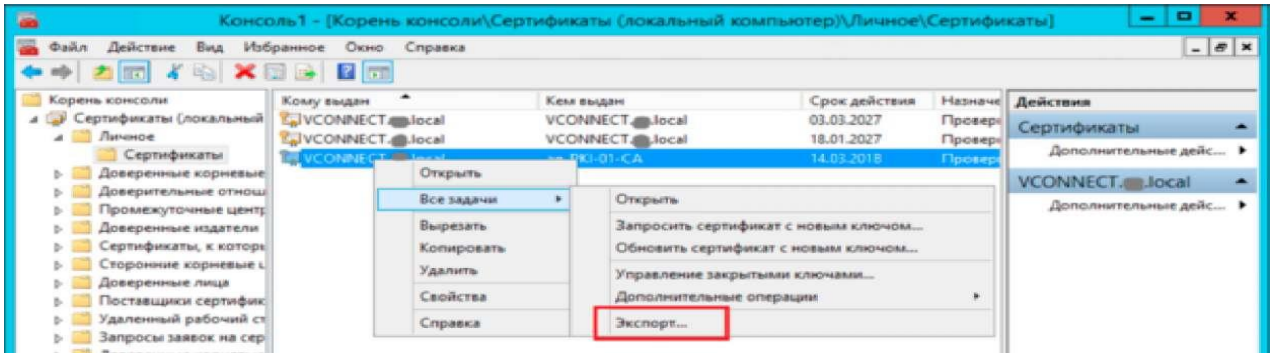


Рисунок 3.52 – Експорт запитаного сертифіката

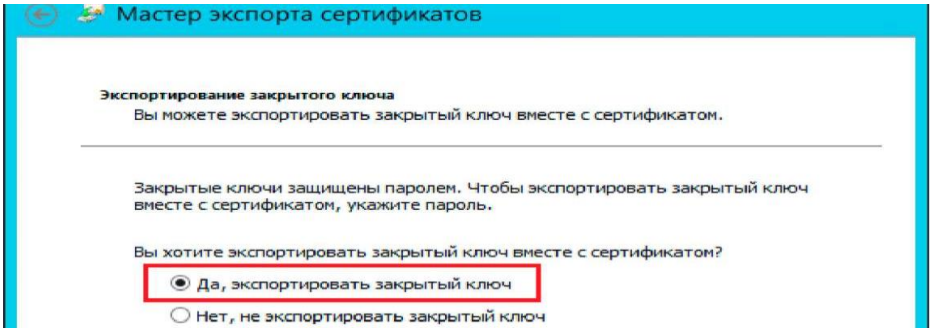


Рисунок 3.53 – Експорт запитаного сертифіката

Всі завдання –> Експорт. Для поновлення сертифіката слід перезапустити службу View Connection Server (рисунок 3.54). Якщо після

перезапуску служби Connection Server не запущені VMware Horizon View Security Gateway Component, це означає, що запитаний сертифікат не доступний для експорту.

Тепер відкриваємо консоль Horizon у браузері і бачимо у властивостях з'єднання, що використовується шифрування (рисунок 3.55), піктограму Connection Server на Dashboard, позначену зеленим, а у властивостях SSL-сертифіката відображується Valid (рисунок 3.56). Зверніть увагу, що консоль Horizon доступна після перезапуску служби View Connection Server не одразу: слід почекати кілька секунд для того, щоб вебсайт був доступний [8–14].

Аналогічно запитується сертифікат і для Security Server (рисунок 3.57), але для того, щоб іконка його статусу стала зеленою, можуть знадобитися додаткові налаштування [8–14].

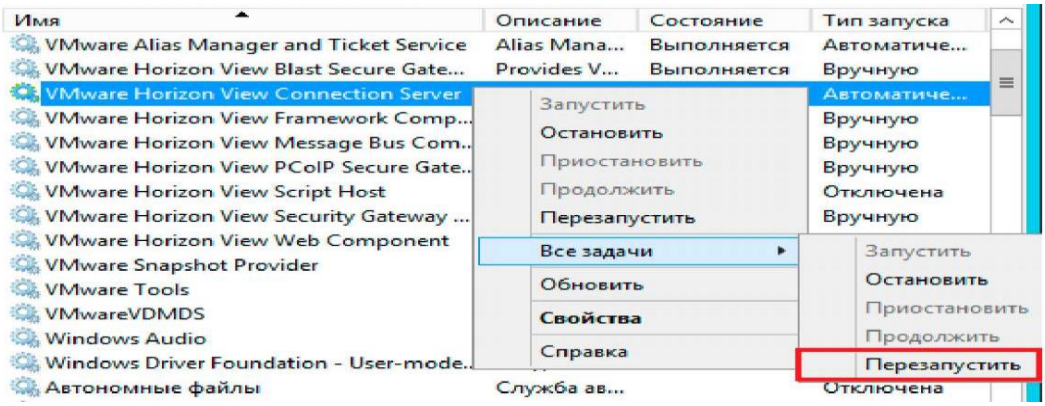


Рисунок 3.54 – Перезапуск служб Connection Server

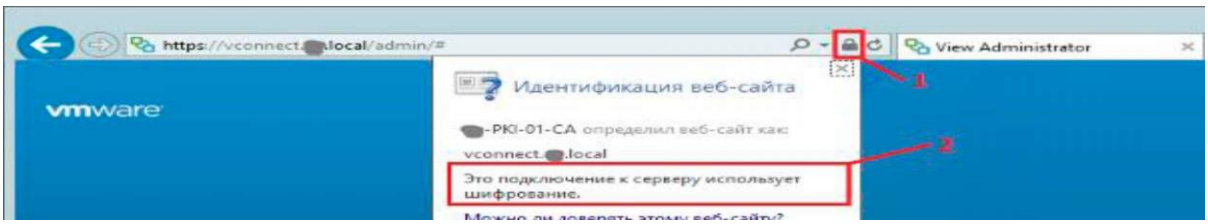


Рисунок 3.55 – Властивості сертифіката Connection Server

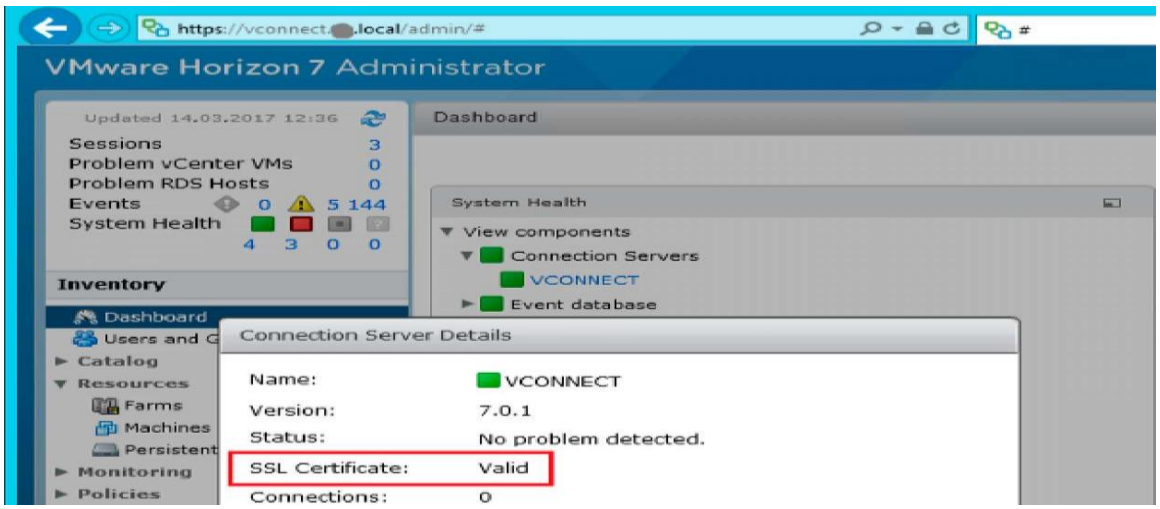


Рисунок 3.56 – Властивості сертифіката Connection Server^

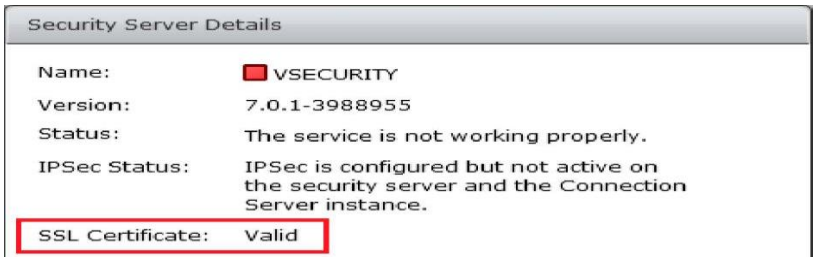


Рисунок 3.57 – Властивості сертифіката

3.2.9 Підключення через Security Server

Security Server – це екземпляр View Connection Server, який надає додатковий рівень безпеки між Інтернетом і вашою внутрішньою мережею. Ви можете встановити один або кілька серверів безпеки для підключення до примірника View Server Connection. Security Server не може бути встановлений на одній віртуальноій або фізичній машині разом з іншими компонентами Horizon View (replica server, View Connection Server, View Composer, View Agent, View Client, or View Transfer Server). За додатковою інформацією можна звернутися до офіційної документації [8–14].

Перш ніж встановити з'єднання через Security Server, необхідно переконатися, що всі налаштування виконані правільно (рисунок 3.58).

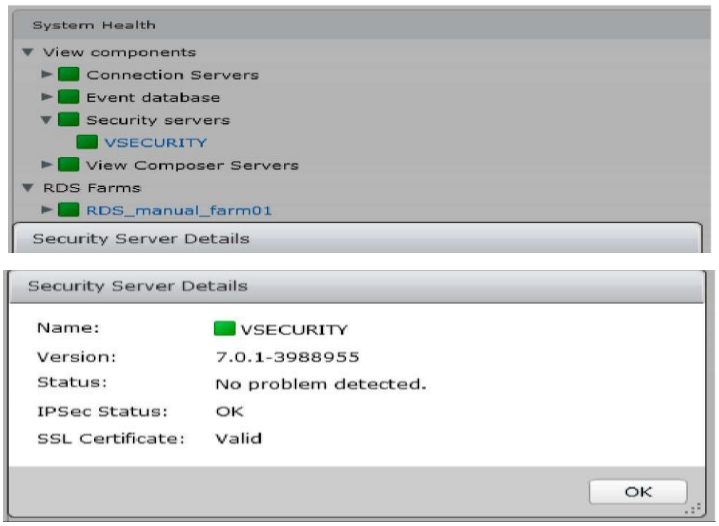


Рисунок 3.58 – Коректно налаштований Security Server

У Horizon View 7 усі налаштування брандмауера і правила безпеки створюються на стадії установлення компонентів, у більш ранніх версіях може знадобитися їх створення. Так само потрібно ввімкнути PColP Secure Geteway і задати зовнішній URL (External URL), відкривши в консолі Horizon (рисунок 3.59): *View Configuration –> Servers –> Connection Servers –> [Connection Server Name] –> Edit –> General.*

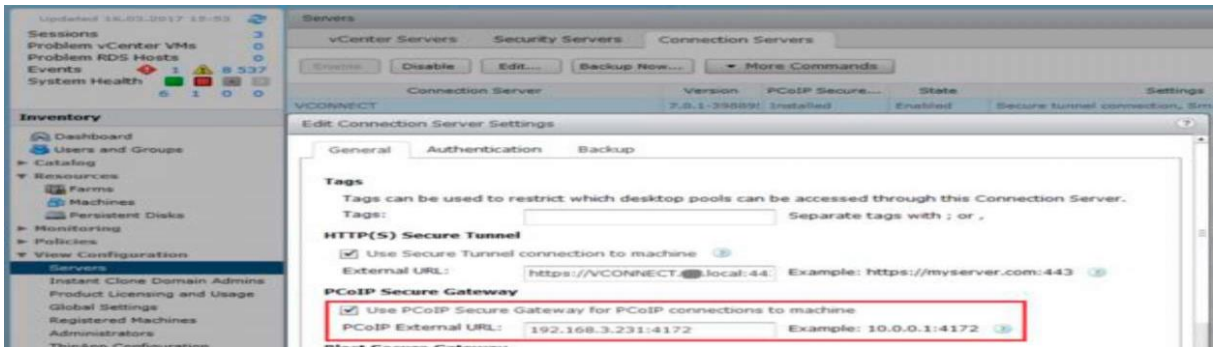


Рисунок 3.59 – Задавання Extenral URL

За замовчуванням, View Connection і Security Server підключаються через тунель, який знаходиться в одній мережі. Клієнти, які працюють за межами цієї мережі, мають підключатися через client–resolvable URL (наприклад 192.168.3.231:4172). За додатковою інформацією можна звернутися до офіційної документації. Тепер запускаємо Horizon View

Client, вказуємо IP-адресу або FQDN Security Server і підключаємося через Secure Gateway (рисунок 3.60).

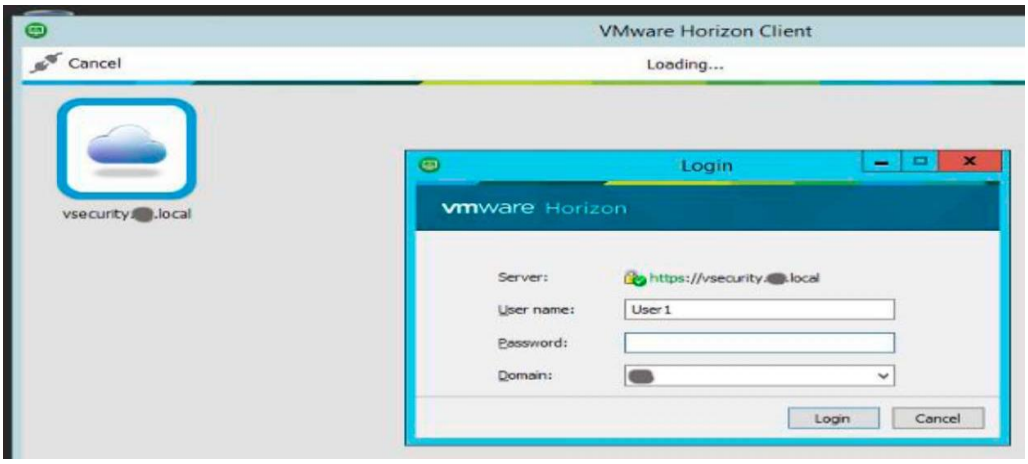


Рисунок 3.60 – Підключення через Security Server

3.3 Порядок виконання лабораторної роботи

1 Установити і розгорнути View.
2 Створити Instant Clone Desktop Pool.
3 Оновити образ Instant Clone.
4 Створити ODBC Database Connection для Composer Server.
5 Створити Linked–Clone Desktop Pool.
6 Порівняти Instant Clones і Linked–Clones.
7 Створити SSL сертифікатів.
8 Установити Microsoft Certificate Authority (ЦС).
9 Підключити через Security Server.

3.4 Зміст звіту

Звіт оформляється кожним здобувачем індивідуально і має містити:
– титульний аркуш з номером і назвою роботи; мету роботи;
– скрини створених View, Instant Clone Desktop Pool;
– скрини оновлених образів Instant Clone;

– скрини створених ODBC Database Connection для Composer Server та Linked-Clone Desktop Pool;

– результати порівняння Instant Clones і Linked-Clones;

– сертифікат SSL;

– скрин встановленого Microsoft Certificate Authority (CA);

– Security Server;

– висновки з роботи.

Контрольні запитання

1 У чому суть віртуалізації серверів?

2 Яке основне завдання функції балансування навантаження в обчислювальних фермах?

3 У чому полягає завдання Composer Server?

4 Виконайте порівняння режимів User Assignment для Linked-Clone Desktop Pool.

5 Які рішення підтримують Instant Clone десктопи?

6 Яке призначення Security Server?

Лабораторна робота 4. Ознайомлення з платформою хмарних обчислень MICROSOFT AZURE

Мета роботи - набуття уміння інсталювати міжмережеві шлюзові рішення для інтеграції інструментної бази хмарних обчислювальних засобів і віртуальної інфраструктури організації.

4.1 Методичні вказівки з організації самостійної роботи здобувачів

Коли компанія переносить свою інфраструктуру у хмарне середовище, то це означає, що частина завдань і відповідальності перекладається на плечі зовнішнього сервіс-провайдера. Завдяки цьому IT-відділ компанії може почати мислити стратегічно, більш грамотно планувати витрати і розвивати бізнесу в необхідному напрямі, не думаючи про підтримку обладнання [1–7].

Якщо бізнесу не доводиться виділяти ресурси на підтримку і забезпечення інфраструктури, то це дає йому перевагу над конкурентами, якщо ті все ще закуповують власне обладнання (а також орендують для серверів площі, оплачують електроенергію). Можливість спрямувати вивільнені бюджетні кошти на розвиток проєкту дає змогу випередити «суперників».

Ситуацію з розвитком хмарних технологій можна порівняти з життям у сучасному мегаполісі. Мешканці багатоквартирних будинків не займаються обслуговуванням водопроводу і електромереж, не ремонтують самостійно під'їзди та ліфти – для всього цього є спеціально навчені люди, які працюють у «зовнішніх» організаціях.

Більш того, доволі очевидно: якщо людина є фахівцем у якійсь одній справі (наприклад професійний сантехнік), то вона виконає роботу якісніше, ніж «і швець, і жнець, і на дуді гравець».

Аналогічна ситуація виникає з хмарними провайдерами. Провайдер має ресурси, які дають йому можливість наймати кращих фахівців, забезпечувати безпеку як даних, так і обладнання та ефективно

використовувати наявну інфраструктуру. Однак у цьому бізнесі критично важливим моментом є довіра клієнтів, яку потрібно заслужити, а після ще й зберегти [1–7].

Недостатньо просто надавати якісні послуги, необхідно також забезпечувати гідне обслуговування клієнтів. Навіть якщо ваша компанія краща в місті, але ви не вмієте працювати з клієнтами, більшого ви не сягнете.

З будь-яким продуктом або послугою періодично виникають проблеми, тому користувачам потрібно кудись звернутися, де їм допоможуть і пояснять, у чому причина. Важливим підходом є створення спеціалізованого посередника, наприклад шлюзу, який вирішує питання інтеграції різних систем [1–7].

4.2 Практична частина

У нашому розпорядженні буде такий сервер для налаштування шлюзу:

Cat / etc / redhat-release.

Використаємо образ minimal для установлення CentOS 7.

На сервері дві мережеві карти eth0 і ethl:

– eth0 підключена до інтернету;

– ethl підключена до локальної мережі разом з комп'ютерами.

У цій лабораторній роботі ми виконаємо необхідні попередні налаштування на сервері, ввімкнено nat, наслаштуємо firewall і встановимо засіб моніторингу мережевої активності.

Якщо у вас недостатньо досвіду і ви не відчуваєте в собі сил розібратися з налаштуванням шлюзу самому за допомогою консолі сервера, спробуйте дистрибутив на основі centos для організації шлюзу і проксі сервера в локальній мережі – clearos. За його допомогою можна через браузер налаштувати весь необхідний функціонал.

Попереднє налаштування сервера [8–14]. Будь-яке налаштування сервера починають з оновлення: *Yum –y update.*

Після цього встановлюють mc: *Yum –y install mc.*

Далі відключаємо selinux.

Знаходимо файл */ etc / sysconfig / selinux* і редагуємо його: *Mcedit / etc / sysconfig / selinux.*

Приводимо рядок з відповідним параметром до такого вигляду: *SELINUX = disabled*.

Щоб застосувати зміни, перезавантажуємо сервер: reboot.

Більш докладно про базове налаштування сервера CentOS 7 йдеться в роботах [8–14].

Тепер налаштуємо мережу. У роботах [8–14] детально розглянуто питання налаштування мережі в CentOS 7, тому виконуємо необхідні команди, без пояснень.

Спочатку видаляємо NetworkManager. Він нам не знадобиться, виконаємо всі налаштування вручну. Іноді він може викликати незрозумілі помилки, краще ним не користуватися:

 # Systemctl stop NetworkManager.service

Systemctl disable NetworkManager.service.

Тепер вмикаємо класичну службу мережі в CentOS 7:

Systemctl enable network.service.

Налаштовуємо мережеві інтерфейси:

Mcedit / etc / sysconfig / network–scripts / ifcfg–eth0

HWADDR = 00: 15: 5D: 01: 0F: 06 TYPE = "Ethernet" BOOTPROTO = "dhcp" DEFROUTE = "yes" PEERDNS = "yes" PEERROUTES = "yes" NAME = "eth0"

UUID = "4e65030c–da90–4fb8–bde4–028424fe3 710 " ONBOOT = "yes"

Mcedit / etc / sysconfig / network–scripts / ifcfg–eth1 DEVICE = eth1

HWADDR = 00: 15: 5d: 01: 0f: 12 TYPE = Ethernet ONBOOT = yes IPADDR = 192.168.10.1 NETMASK = 255.255.255.0.

Перезапускаємо службу мережі: Systemctl restart network.service та дивимося на результат: **Ipa** (рисунок 4.1).

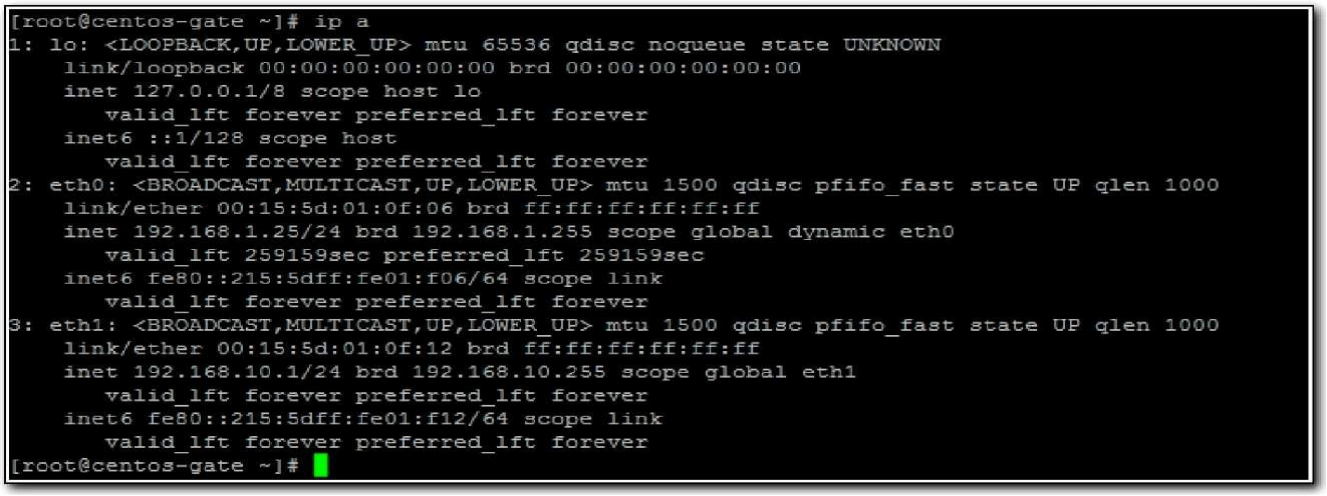


Рисунок 4.1 – Результат налаштування мережевих інтерфейсів

Ви налаштовуєте мережу залежно від своїх умов. Якщо зовнішній адаптер отримує налаштування не по dhcp, а в статиці, то не забудьте налаштувати шлюз за замовчуванням і dns сервер.

Перш ніж рухатися далі, переконайтеся, що ви все правильно налаштували: на сервері працює інтернет, комп'ютери з локальної мережі, пінг сервер за адресою на ethl.

Вмикаємо маршрутизацію: firewall і nat.

Щоб сервер міг маршрутизувати пакети між мережевими адаптерами, необхідно виконати таке налаштування. Знаходимо файл */etc/sysctl.conf* і вставляємо туди рядок:

Mcedit /etc/sysctl.conf, net.ipv4.ipforward = 1.

Щоб запрацювало налаштування, виконуємо команду Sysctl-p.

Тепер приступаємо до найголовнішого – налаштування фаєрвола. У роботах [8–14] дуже докладно розглянуто питання налаштування iptables в CentOS 7, там же наведено готовий скрипт для iptables. Тут виконуємо всі необхідні дії без пояснень [8–14].

Відключаємо firewalld:

Systemctl stop firewalld, Systemctl disable firewalld.

Встановлюємо служби iptables:

Yum – y install iptables–services.

Викачуємо скрипт з правилами iptables.sh. Ці правила включають NAT, закривають доступ до сервера зовні, дають змогу пінг, а всім

користувачам локальної мережі доступ в інтернет. Додатковий функціонал відключений. У скрипті докладно описані всі правила. Вам необхідно тільки замінити на початку змінні на свої. У цьому випадку це буде виглядати так [8–14].

Зовнішній інтерфейс export WAN = eth0

export WAN IP = 192.168.1.25.

Локальна мережа export LAN1 = eth1

export LAN1IPRANGE = 192.168.10.1 / 24.

Розміщуємо відредагований скрипт в */ etc / iptabes.sh* і робимо його виконуваним:

Chmod 0740 /etc/iptables.sh.

Запускаємо iptables:

Systemctl start iptables.service.

Додаємо їх в автозавантаження:

Systemctl enable iptables.service .

Виконуємо скрипт з правилами:

/etc/iptables.sh.

Перевіряємо встановлені правила:

Iptables –L –v –n.

Якщо у вас те саме, то ви все зробили правильно. По суті шлюз вже готовий і може обслуговувати клієнтів. Але не працює одна важлива служба, без якої нормальної роботи з інтернетом не вийде. Потрібно налаштувати dns, який кешує сервер для клієнтів локальної мережі. Можна піти двома шляхами. Перший – це виконати найпростіше налаштування dns сервера bind [8–14]. Другий – це встановити dnsmasq, який, крім dns сервера, включає ще й dhcp сервер, який нам може стати в нагоді.

Установлення і налаштування dnsmasq в CentOS 7. З великою ймовірністю dnsmasq у вас уже встановлений. Перевірити це можна за допомогою команди *Rpm –qa | grep dnsmasq dnsmasq–2.66–14.el 7_1.x86_64* (рисунок 4.2).

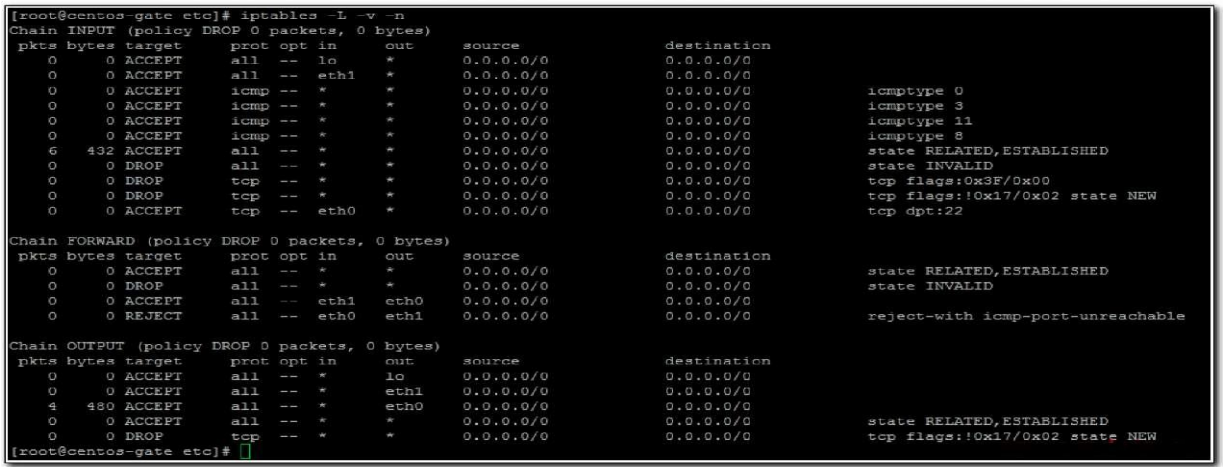


Рисунок 4.2 – Проміжні налаштування

Якщо висновок такий самий, то пакет уже стоїть. Якщо ні, то встановлюємо dnsmasq командою

Yum –y install dnsmasq.

Редагуємо файл конфігурації */etc/dnsmasq.conf* і приводимо його до простого вигляду:

Mcedit /etc/dnsmasq.conf

domain–needed

bogus–priv

interface = eth1

dhcp–range = 192.168.10.50,192.168.10.150,24h.

Запускаємо dnsmasq:

Systemctl start dnsmasq

або перезапускаємо, якщо він був у вас запущений:

Systemctl restart dnsmasq.

Додаємо dnsmasq в автозавантаження:

Systemctl enable dnsmasq.

На цьому налаштування завершено, шлюзом під CentOS 7 можна користуватися.

Аналіз мережевої активності на шлюзі в Linux. У цій конфігурації шлюз може успішно функціонувати. Але іноді хочеться подивитися, а що взагалі на ньому відбувається. Наприклад, хтось займає весь канал, інтернет гальмує, а ми, як сліпі кошенята, сидимо і не бачимо нічого.

Потрібен якийсь засіб для перегляду завантаження мережі на шлюзі. І такий засіб є – це програма iftop [8–14].

Вона відсутня в стандартному репозиторії CentOS 7. Для її установлення необхідно підключити репозиторій epel:

Yum –y install epel–release.

Встановлюємо iftop на CentOS 7:

Yum –y install iftop.

Тепер ми можемо дивитися завантаження мережі на шлюзі в режимі реального часу. Щоб побачити мережеву активність, досить запустити iftop: iftop.

За замовчуванням, вона слухає зовнішній інтерфейс шлюзу eth0, на ньому всі підключення будуть відображені від імені самого шлюзу і визначити, хто ж у мережі займає канал, ми не зможемо. Щоб це побачити, необхідно запустити перегляд мережевої активності на локальному інтерфейсі. Зробити це нескладно, достатньо запустити iftop з параметром [8–14]:

Iftop – i eth1–P.

Додамо параметр P, що відображує порти, через які проходять з'єднання. Подивимося, хто більше всіх завантажує канал інтернету.

У нашому випадку це користувач з IP 192.168.10.98, на якому ми запустили перевірку швидкості інтернету з серверів google.

Якщо у вас невелика мережа і небагато користувачів, то за допомогою цієї простої і ефективної утиліти ви зможете легко визначити, хто, наприклад, хитає торенти або чимось ще завантажує канал [8–14].

4.3 Порядок виконання лабораторної роботи

1 Установити платформу хмарних обчислень MICROSOFT AZURE.

2 Розгорнути платформу хмарних обчислень MICROSOFT AZURE

3 Вивчити сервер для налаштування шлюзу.

4 Налаштувати мережеві інтерфейси.

5 Налаштувати шлюз за допомогою консолі сервера.

6 Налаштувати мережу. Спочатку видалити NetworkManager.

4.4 Зміст звіту

Звіт оформляється кожним здобувачем індивідуально і має містити:

– титульний аркуш з номером і назвою роботи;

– мету роботи;

– умовне графічне зображення досліджуваних тригерів;

– VHDL-опис тригерів;

– часові діаграми перевірки правильності роботи тригерів;

– лістинги UCF файлів;

– результати автоматизованого синтезу проєкту у вигляді графічного відображення схем тригерів;

– висновки з роботи.

Звіт має складатися зі скриншотів налаштувань та обґрунтування (15–20 сторінок) щодо продуктивності використання.

Звіт готується один на бригаду. Матеріали п. 4.1–4.4 до звіту не включаються.

Контрольні запитання

1 Який тригер називається асинхронним і синхронним?

2 Чим відрізняється синхронізація за рівнем від синхронізації за фронтом?

3 У чому відмінність між асинхронним і синхронним встановленням тригерів у початковий стан?

4 Що таке latch і flip–flop тригери?

5 У чому полягає особливість MS-структури тригерів?

6 Які конструкції VHDL використовуються для опису синтезованих моделей асинхронних тригерів, тригерів, синхронізованих рівнем, і тригерів, синхронізованих фронтом?

Рекомендації щодо організації самостійної роботи

Самостійна робота здобувача вищої освіти — це форма організації освітнього процесу, при якій заплановані завдання виконуються здобувачем під методичним керівництвом викладача, але без його безпосередньої участі. Самостійна робота здобувача є основним засобом засвоєння навчального матеріалу під час позааудиторної навчальної роботи. Самостійна робота спрямована на закріплення теоретичних знань, отриманих здобувачами за час навчання, поглиблення і удосконалення.

Метою самостійної роботи є системне і послідовне формування компетентностей майбутнього фахівця, досягнення очікуваних результатів навчання та формування у здобувачів вищої освіти самостійності у здобутті і поглибленні знань, що сприятиме підвищенню конкурентоспроможності майбутніх фахівців на сучасному ринку праці.

Основними завданнями самостійної роботи є послідовне вироблення навичок ефективної самостійної професійної (практичної й науково-теоретичної) діяльності на рівні європейських і світових стандартів.

Зміст самостійної роботи здобувача освіти визначається робочою навчальною програмою дисципліни, завданнями та рекомендаціями викладача.

Організаційні форми самостійної роботи здобувачів:

– робота виконується самостійно поза аудиторією або з урахуванням специфіки дисципліни в лабораторії, спеціалізованій аудиторії чи майстерні закладу;

– індивідуальна робота здійснюється за персоналізованим завданням під керівництвом викладача, під час виконання якої здобувач може отримати методичну допомогу у вигляді індивідуальної консультації. Така робота може включати вивчення окремих розділів навчальної дисципліни, виконання творчої роботи, спортивне тренування, роботу з використанням комп’ютерної техніки або лабораторного обладнання тощо. Ці завдання можуть бути навчального, навчально-дослідного, творчого, проєктно-конструкторського характеру тощо. Головна мета - поглиблення, узагальнення та закріплення знань, які здобувачі набувають у процесі навчання, а також застосування цих знань на практиці;

– індивідуальні завдання видаються здобувачам у терміни, передбачені робочою навчальною програмою дисципліни, і виконуються кожним здобувачем самостійно при консультуванні з викладачем. У випадках, коли завдання мають комплексний характер, до їх виконання залучаються декілька здобувачів, у тому числі інших спеціальностей.

На самостійну роботу можуть виноситися:

– частина теоретичного матеріалу, менш складного за змістом;

– окремі практичні роботи, які не потребують безпосереднього керівництва викладача;

– окремі питання, що висвітлюються в літературних джерелах і не розглядаються на лекціях;

– підготовка до семінарських, практичних (лабораторних) занять;

– вирішення та письмове оформлення задач, схем, діаграм, інших робіт графічного характеру;

– відпрацювання тренінгових програм (завдань) з навчальних дисциплін;

– аналіз конкретної виробничої ситуації та підготовка аналітичної записки;

– підготовка практикуму з навчальної дисципліни з використанням програмного забезпечення;

– виконання індивідуальних завдань (написання реферату за заданою проблематикою);

– пошук (підбір) та огляд літературних джерел за заданою проблематикою курсу;

– аналітичний розгляд наукової публікації;

– написання курсового/дипломного проєкту (роботи) тощо.

Самостійна робота здобувачів освіти включає:

– підготовку до аудиторних занять (лекцій, практичних, лабораторних тощо);

– виконання завдань з навчальної дисципліни протягом семестру;

– роботу над окремими темами навчальних дисциплін, які згідно з робочою навчальною програмою дисципліни винесені на самостійне опрацювання;

– підготовку до всіх видів контрольних випробувань, у тому числі курсових, модульних і комплексних контрольних робіт;

– виконання завдань, передбачених програмою практики;

– роботу в студентських наукових гуртках, семінарах тощо;

– участь у роботі факультативів, спецсемінарів тощо;

– участь у наукових і науково-практичних конференціях, семінарах, конкурсах, олімпіадах тощо;

– підготовку до підсумкової державної атестації, у тому числі виконання випускної кваліфікаційної роботи відповідного освітньо-кваліфікаційного рівня та освітньо-професійного ступеня.

Для організації самостійної роботи здобувачів необхідні такі умови:

– готовність здобувачів до самостійної роботи, наявність мотивації до надбання знань;

– наявність і доступність необхідної навчально-методичної літератури та довідкового матеріалу;

– наявність відповідної матеріально-технічної бази: навчально-лабораторного обладнання, комп`ютерних класів тощо;

– система регулярного контролю якості виконаної самостійної роботи;

– консультаційна допомога викладача.

Самостійна робота здобувачів забезпечується системою навчально-методичних засобів, передбачених для вивчення конкретної навчальної дисципліни:

– основна література (підручник, навчальні та методичні посібники, конспект лекцій викладача, практикум тощо);

– додаткова література (наукова, фахова, монографічна, періодична);

- методичні матеріали, що мають передбачити можливість проведення самоконтролю з боку здобувача освіти.

Самостійна робота, яка не передбачена освітньо-професійною програмою (навчальним планом і навчально-методичними матеріалами), але сприяє повнішому розкриттю і конкретизації її змісту, може здійснюватися з ініціативи здобувача з метою реалізації його власних навчальних і наукових інтересів.

Список літератури

1 Лістровий С. В., Мірошник М. А. Інформаційно-управляючі системи та організація паралельних обчислювань: навч. посіб. Харків: «Діса плюс», 2015. 324 с.

2 Tulloch M. Uderstanding MS Virtualization Solutions.2 ed. 2015. 320 p.

3 Bessoudo J. Using SUN Systems to Build a Virtual and Dynamic Infrastructure. SUN. 2015. 110 p.

4 Agesen O. Software Techniques for Avoiding Hardware Virtualization Exits. VMware. 2014. 620 p.

5 Muller A.Virtualization with VMware ESX Server. Syngress. 2014. 384 p.

6 Ventresco J. VMware Horizon View 6.0 Desktop Virtualization Cookbook. Packt. 2015. 228 p.

7 Жилін А. В., Шаповал О. М., Успенський О. А. Технології захисту інформації в інформаційно-телекомунікаційних системах: навч. посіб. Київ: КПІ ім. Ігоря Сікорського; Вид-во «Політехніка», 2021. 213 с.

8 Романов О. І., Нестеренко М. М., Фесьоха Н. О. Аналіз сучасних технологій віртуалізації для побудови інформаційно-телекомунікаційних систем. *Збірник наукових праць Військового інституту телекомунікацій та інформатизації.* 2019. Вип. 1. С. 82-90.

9 Костюченко А. О., Горошко Ю. В. Віртуалізація операційних систем: навч.-метод. посіб. Чернігів: ФОП Баликіна С. М., 2021. 56 с.

10 Мірошник М. А. Комп’ютерні технології автоматизованого проектування: навч. посіб. Харків: ХНУРЕ, 2007. 300 с.

11 Мірошник М. А. Конспект лекцій з дисциплін «САПР пристроїв і систем автоматики» та «Основи систем автоматизації проектування» для студентів спеціальності 123 СКС. Харків: УкрДУЗТ, 2013. 102 с.

12 Лістровий С. В., Мірошник М. А. Теорія автоматичного управління, штучний інтелект і автоматизація процесу прийняття рішення: навч. посіб. Харків: УкрДУЗТ, 2018. 144 с.

13 Мірошник М. А. Проектування діагностичної інфраструктури обчислювальних систем та пристроїв на ПЛІС: монографія. Харків: ХУПС, 2012. 188 с.

14 Леонов С. Ю., Загарій Г. І. Автоматизоване проектування складних систем у комп'ютерній схемотехніці: навч. посіб. Харків: ПП видавництво «Нове слово», 2012. 287 с.

МЕТОДИЧНІ ВКАЗІВКИ

до лабораторних робіт і самостійної роботи з дисципліни

«*ПАРАЛЕЛЬНІ ТА РОЗПОДІЛЕНІ ОБЧИСЛЕННЯ ТА CKLOUD-ТЕХНОЛОГІЇ*»

Відповідальний за випуск Мірошник М. А.

Редактор Ібрагімова Н. В.

Підписано до друку 30.06.2021 р.
Умовн. друк. арк. 8,0. Тираж . Замовлення № .

Видавець та виготовлювач Український державний
університет залізничного транспорту,
61050, Харків-50, майдан Фейєрбаха,7.
Свідоцтво суб'єкта видавничої справи ДК № 6100 від 21.03.2018 р.