

**УКРАЇНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
ЗАЛІЗНИЧНОГО ТРАНСПОРТУ**

**ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КЕРУЮЧИХ СИСТЕМ ТА  
ТЕХНОЛОГІЙ**

**Кафедра спеціалізованих комп'ютерних систем**

**МЕТОДИЧНІ ВКАЗІВКИ**

**до практичних занять і розрахунково-графічної роботи з дисциплін**

***«ПАРАЛЕЛЬНІ ТА РОЗПОДІЛЕНІ ОБЧИСЛЕННЯ  
ТА SKLOUD-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 SKLOUD-технологій.

Методичні вказівки призначено для здобувачів вищої освіти напрямку 123 «Комп'ютерна інженерія», які вивчають дисципліну «Паралельні та розподілені обчислення та SKLOUD технології», денної та заочної форм навчання.

Укладачі:

проф. М. А. Мірошник,  
асист. Ю. В. Савін

Рецензент

доц. С. І. Доценко

## ЗМІСТ

Вступ	4
Практичне заняття 1. Ознайомлення з віртуалізацією. Система віртуалізації ORACLE VM VIRTUALBOX	6
Практичне заняття 2. Встановлення і використання VMWare WorkStation. Зміни конфігурації віртуальної машини за допомогою VMWare WorkStation	15
Практичне заняття 3. Створення, вивчення та використання вебдодатка WORD PRESS за допомогою хмарного сервісу MICROSOFT AZURE	27
Практичне заняття 4. Архітектура і процес автоматизованого розгортання віртуальних машин. Створення автошаблону віртуальних машин	40
Практичне заняття 5. Створення файлів визначення та завдань. Розгортання віртуальної машини	48
Практичне заняття 6. Модель обслуговування SAAS. Розгортання хмарного сховища OWNCKLOUD	54
Практичне заняття 7. Модель обслуговування PAAS на прикладі деплоя додатка в HEROKU	59
Практичне заняття 8. Використання моделі обслуговування PAAS для виконання розрахунково-графічної роботи	61
Список літератури	63
Додаток А Створення віртуальної машини у VirtualBox	65
Додаток Б Встановлення операційної системи Ubuntu у VirtualBox	70
Додаток В Встановлення Debian GNU / Linux у VirtualBox	73
Додаток Г Встановлення ownCkcloud Server у Debian GNU/Linux	86
Додаток Д Підключення ownCkcloud Client до сервера	89
Додаток Е Підключення та реєстрація до Heroku	90

## ВСТУП

Методичні вказівки призначені для виконання практичних занять з дисципліни «Паралельні та розподілені обчислення та Scloud-технології» (ПРО Scloud T), введеної в навчальний процес на підставі навчальних планів за вищевказаним напрямом. Метою викладання дисципліни є отримання знань з принципів побудови та функціонування інформаційних систем за допомогою Scloud-технологій.

Методичні вказівки містять опис восьми практичних занять. Кожне практичне завдання містить такі підрозділи: мета заняття, теоретична частина, вказівки з підготовки до виконання практичного заняття, зміст звіту, контрольні запитання.

За результатами вивчення дисципліни здобувачі вищої освіти повинні знати:

- принципи організації сучасних технологій віртуалізації та призначення, основні принципи функціонування систем на базі технологій віртуалізації;
- різновиди технологій віртуалізації;
- основні напрями розвитку в галузі технологій віртуалізації та концептуальні перспективи їхнього розвитку.

Також здобувачі вищої освіти повинні вміти розробляти архітектуру комп'ютерної системи з використанням scloud-технологій; обирати scloud-технологію відповідно до сфери застосування; проектувати комп'ютерні мережі у scloud-середовищі; інтегрувати scloud-технології в існуючі комп'ютерні системи; експлуатувати комп'ютерні системи, побудовані з використанням scloud-технологій; створювати програмні додатки для scloud-середовища.

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

Вивчення дисципліни «Паралельні та розподілені обчислення та Scloud-технології» закінчується розрахунково-графічною роботою та іспитом. При цьому виді контролю іспит вважається складеним успішно, якщо протягом семестру здобувач вищої освіти 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. Відпрацювати і захистити всі практичні заняття і виконати і захистити індивідуальну розрахунково-графічну роботу.

## **Практичне заняття 1. Ознайомлення з віртуалізацією. Система віртуалізації ORACLE VM VIRTUALBOX**

**Мета роботи:** ознайомитися з основами віртуалізації, системою віртуалізації *VirtualBox*, навчитися налаштовувати віртуальну машину (ВМ), здійснювати прості операції з нею, встановлювати гостьову операційну систему на ВМ.

### **1.1 Теоретичні основи віртуалізації**

В обчисленнях віртуалізація є процесом створення віртуальної (не фізичної) версії чого-небудь, зокрема апаратних платформ віртуального комп'ютера, операційних систем, пристроїв зберігання даних і обчислювальних ресурсів [1–7].

Віртуалізація може бути надана на різних апаратних і програмних рівнях: центральний процесор, диск, пам'ять, файлові системи та ін. Найчастіше віртуалізація використовується для створення ВМ і емуляції різного устаткування для подальшого встановлення операційних систем (ОС) на них [8–14]. Віртуалізацію можна використовувати:

– у консолідації серверів (можливість мігрувати фізичні сервери на віртуальні, у такий спосіб збільшується коефіцієнт використання серверних потужностей, що дає змогу істотно заощадити на апаратурі, електроенергії та обслуговуванні);

– розробленні та тестуванні додатків (можливість одночасно запускати кілька різних ОС, що зручно при розробленні крос-платформного програмного забезпечення (ПЗ), у такий спосіб значно підвищується якість, швидкість розроблення і тестування додатків);

– бізнесі (використання віртуалізації в бізнесі зростає з кожним днем і постійно знаходяться нові способи застосування цієї технології, наприклад можливість безболісно зробити снапшот і швидко відновити систему при збої);

– організації віртуальних робочих станцій (так званих «тонких клієнтів»).

Загальна схема взаємодії віртуалізації з апаратним і програмним забезпеченням (ПЗ) подана на рисунку 1.1.

Взаємодія додатків і ОС з апаратним забезпеченням здійснюється через абстраговані шари віртуалізації [1–7].

Існує кілька підходів організації віртуалізації: емуляція обладнання (QEMU, Vochs, Dynamips); повна віртуалізація (KVM, HyperV, VirtualBox); паравіртуалізація (Xen, L4, Trango); віртуалізація рівня ОС (LXC, OpenVZ, Jails, Solaris Zones).

Емуляція апаратних засобів є одним з найскладніших методів віртуалізації (рисунок 1.2). Водночас головною проблемою при емуляції апаратних засобів залишається низька швидкість роботи, причиною чого є те, що кожна команда моделюється на основних апаратних засобах [8–14].



Рисунок 1.1 – Схема взаємодії віртуалізації з апаратурою та ПЗ



Рисунок 1.2 – Емуляція апаратних засобів

В емуляції обладнання використовується механізм динамічної трансляції, тобто кожна з інструкцій емулірованої платформи заміняється на заздалегідь підготовлений фрагмент інструкцій фізичного процесора.

При повній віртуалізації поверх уже встановленої ОС встановлюється програма-гіпервізор, яка здійснює взаємозв'язок між гостьовими ОС і хост-комп'ютером (рисунок 1.3).

Перевагою технології повної віртуалізації є можливість устанавлення різних ОС, а недоліком – менша продуктивність внаслідок накладних витрат на гіпервізор, а також зниження швидкості роботи з підсистемою введення/виведення через необхідність ізоляції [1–7].

Паравіртуалізація має деяку схожість з повною віртуалізацією. У цьому методі також використовується гіпервізор для поділу доступу до апаратури, але об'єднується код, що стосується віртуалізації в операційну систему (рисунок 1.4).

Недоліком паравіртуалізації є необхідність зміни гостьової ОС для гіпервізора, однак у такий спосіб набагато збільшується продуктивність [8–14].

Віртуалізація рівня ОС не потребує гіпервізора. Для її роботи необхідно модифіковане ядро на хост-системі з набором патчів і утиліт для управління контейнерами (рисунок 1.5).



Рисунок 1.3 – Повна віртуалізація





Рисунок 1.4 – Паравіртуалізація



Рисунок 1.5 – Віртуалізація рівня ОС

Унаслідок безпосередньої взаємодії контейнера з ядром, без гіпервизора, забезпечується максимальна швидкодія. Але оскільки для всіх контейнерів використовується загальне ядро, то нема можливості використовувати різні ОС в контейнерах. Oracle VM VirtualBox – система віртуалізації, розроблена компанією Innotek у 2007 році, пізніше придбана компанією Sun Microsystems [1–7].

Ключовими можливостями системи є кросплатформеність, наявність графічного інтерфейсу, локалізація, підтримка апаратної віртуалізації, експериментальне VD-прискорення, підтримка різних образів жорстких дисків, можливість установлення доповнень гостьової ОС, наприклад для коректної роботи існуючих USB-пристроїв або можливості зміни розділення робочого столу гостьової ОС [8–14].

## 1.2 Порядок виконання роботи

Хід роботи:

- 1) установити віртуальну машину VirtualBox на комп'ютер (додаток А);
- 2) установити UBUNTU у VirtualBox на VirtualBox (додаток Б).

### 1.2.1 Створення віртуальної машини за допомогою Oracle VM VirtualBox

1 Установити VirtualBox на ПК. На рисунку 1.6 наведено вікна інсталятора Oracle VM VirtualBox.

2 Запустити програму та натиснути «Створити», як показано на рисунку 1.6.

3 Завантажити образ операційної системи, яку намагаєтеся встановити на створену ВМ (рисунок 1.7).

4 Задати ім'я, тип і версію ВМ, що створюється (рисунок 1.8). Для створення ВМ необхідно виділити ресурси фізичного комп'ютера у вигляді частки оперативної пам'яті та простору на жорсткому диску.

5 Створити віртуальний жорсткий диск, на якому система буде зберігати файли (рисунок 1.8). Віртуальна машина успішно створена та з'явиться в переліку віртуальних машин (рисунок 1.9).



Рисунок 1.6 – Вікно інсталятора та привітання Oracle VM VirtualBox



Рисунок 1.7 – Образ ОС

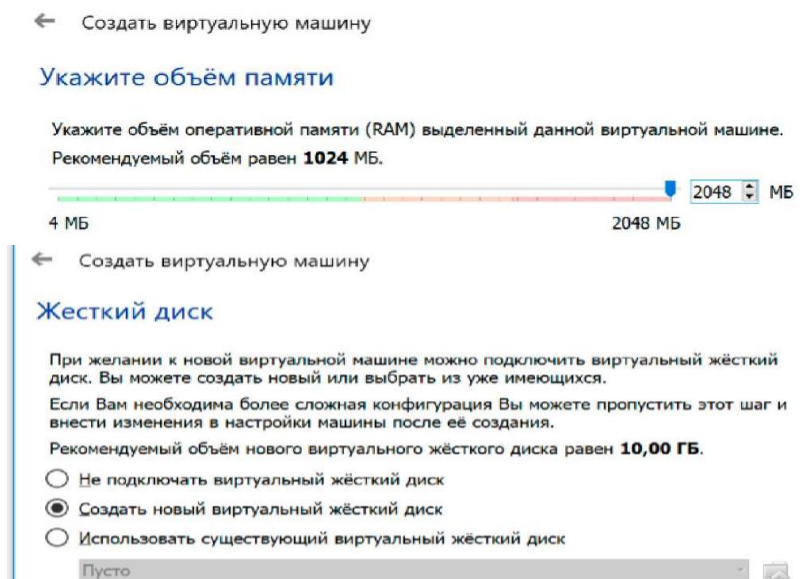


Рисунок 1.8 – Вибір пам'яті та жорсткого диска для віртуальної машини

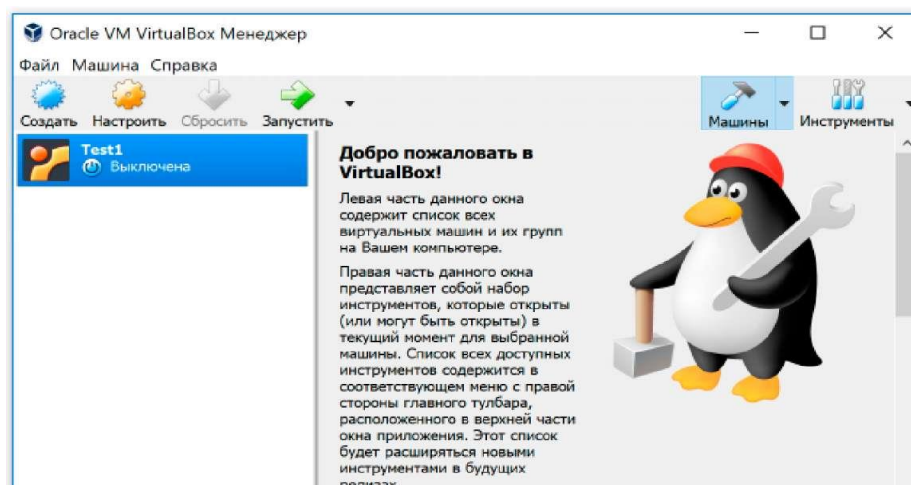


Рисунок 1.9 – Створена VM

## 1.2.2 Запуск віртуальної ОС

Система запросить шлях до образу операційної системи, яку ви вибрали для створеної ВМ (рисунок 1.10). Після цього ВМ готова до роботи.

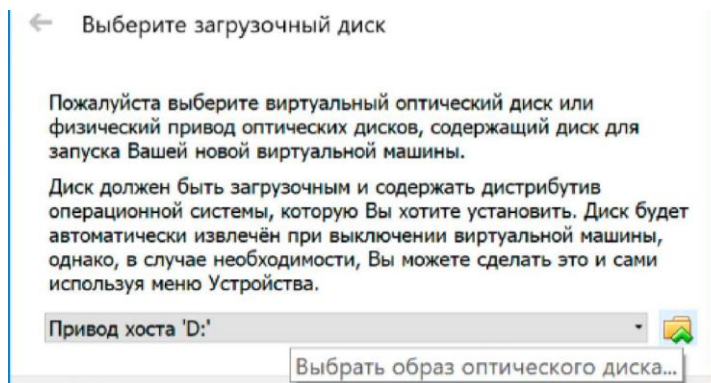


Рисунок 1.10 – Запуск ОС

## 1.2.3 Клонування віртуальних машин

Функціонал щодо клонування та снешотів є дуже необхідним інструментом під час роботі з ВМ.

Щоб не видаляти ВМ через різні обставини, наприклад встановлення підозрілого додатка, і знов не створювати ВМ та налаштовувати систему, можна застосувати два варіанти. Перший – створити зразкову машину і потім клонувати її. Другий - подальше використання ВМ – це створення снешоту або зліпка.

1 Обираєте «Клонувати» і назву клону (рисунок 1.11).

2 Після закінчення процедури обов'язково обираєте повне клонування, і система готова до запуску (рисунок 1.11).

3 Задля вирішення певних завдань запускаєте клон, а після закінчення користування видаляєте його і створюєте новий клон.

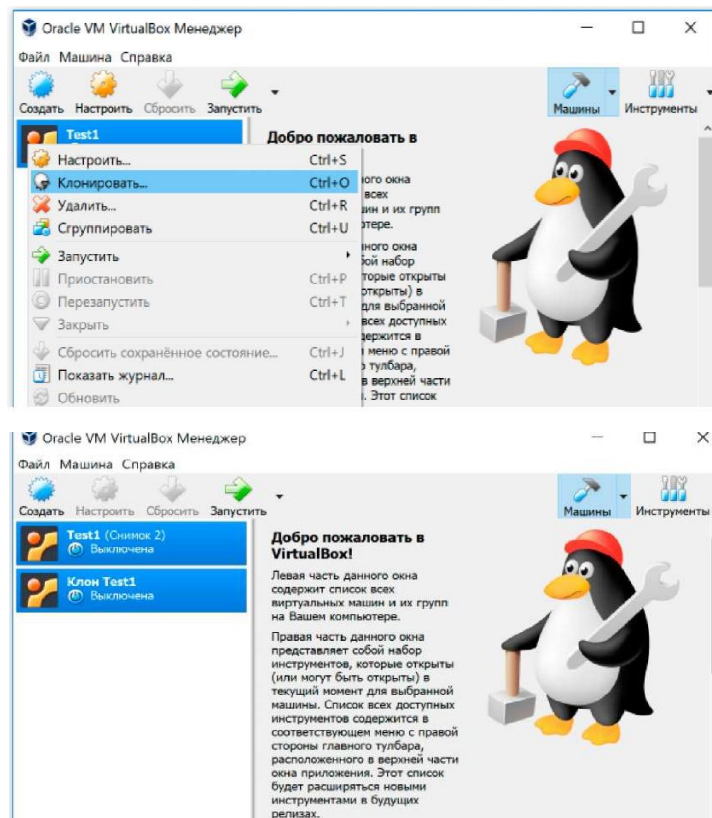


Рисунок 1.11 – Створений клон віртуальної машини

### 1.2.4 Снепшоти віртуальних машин. Снепшот (Snapshot) – це копіювання стану системи, до якого можна звернутися

1 Для створення снепшоту необхідно в меню VirtualBox обрати Машини→Зліпка, а потім «Створити» (рисунок 1.12).

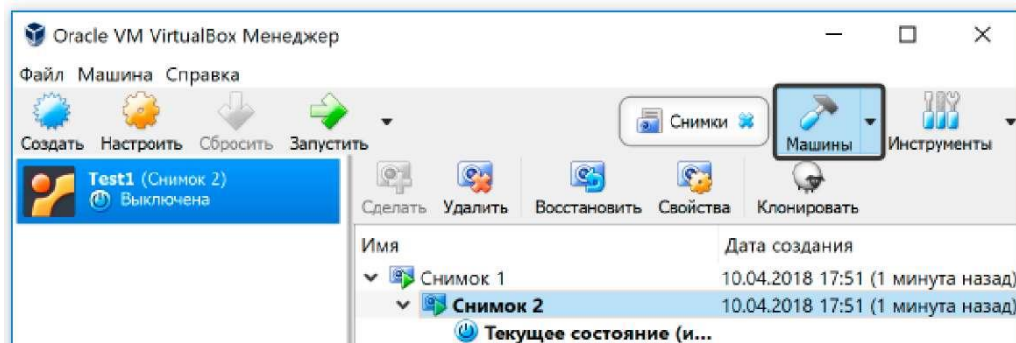


Рисунок 1.12 – Створення снепшоту

2 Кожному снєпшоту необхідно присвоїти ім'я та опис.

3 За необхідності відновлення стану системи треба вибрати версію снєпшоту і натиснути «Відновити».

### **1.3 Зміст звіту**

Після закінчення практичного заняття кожен здобувач повинен скласти звіт, що складається з титульного аркуша, мети заняття, ходу роботи з Prnt Scrn, зроблених завдань і висновків про виконану роботу. Звіт оформляється кожним здобувачем індивідуально.

### **Контрольні запитання**

1 Особливості розподілу задач і передачі даних.

2 Розподілені обчислення (Grid computing).

3 Розподілені інформаційні системи.

4 Типи та особливості розподілених інформаційних систем.

5 Історія виникнення та еволюція skloud-систем.

6 Класифікація skloud-систем.

7 Основні вимоги та характеристики skloud-системи.

8 Відмінності від традиційних розподілених систем.

## **Практичне заняття 2. Встановлення та використання VMWare WorkStation. Зміни конфігурації віртуальної машини за допомогою VMware WorkStation**

**Мета роботи:** практичне засвоєння технології віртуалізації за допомогою VMWare Workstation.

**Апаратне і програмне забезпечення:** настільний або портативний комп'ютер, який підтримує віртуалізацію, встановлене VMWare Workstation, створена віртуальна машина з гостьовою ОС Windows 8.

**Завдання:** використовуючи інсталяційний дистрибутив VMWare Workstation, встановити ПЗ на комп'ютер; задати конфігурацію віртуальної мережі (ВМ); встановити інструменти VMWare Workstation, створити і використати Snapshot; збільшити оперативну пам'ять (ОП) і кількість процесорів; створити додатковий жорсткий диск і новий віртуальний жорсткий диск.

### **2.1 Теоретичні відомості про VMWare WorkStation**

VMware Workstation створює повністю ізольовані безпечні ВМ, що інкапсулюють ОС і додатки. Рівень віртуалізації VMware зіставляє фізичні апаратні ресурси з ресурсами ВМ [1–7]. Отже, кожна ВМ має свій власний процесор, пам'ять, диски та пристрої введення/виведення і є повним еквівалентом стандартного комп'ютера x86. VMware Workstation інсталюється до операційної системи хоста і забезпечує широку підтримку обладнання за рахунок спадкування апаратних засобів ОС хоста [8–14].

## 2.2 Порядок виконання роботи

### 2.2.1 Установлення VMWare Workstation

1 Установити VMWare Workstation. На рисунку 2.1 подано VMware Workstation, вікна інсталятора і ліцензійна угода.

2 Вибрати тип налаштування VMware Workstation і визначити теку призначення (рисунок 2.2). Оновлення та поліпшення ПЗ, створення ярликів, виконання необхідних операцій і запрошеної операції, закінчення налаштування (рисунок 2.3-2.5).

3 Установлення ПЗ на комп'ютер, використовуючи інсталяційний дистрибутив VMware Workstation (рисунок 2.6). Запустити програму, відкрити Пуск → Усі програми → VMWare → VMWare Workstation.

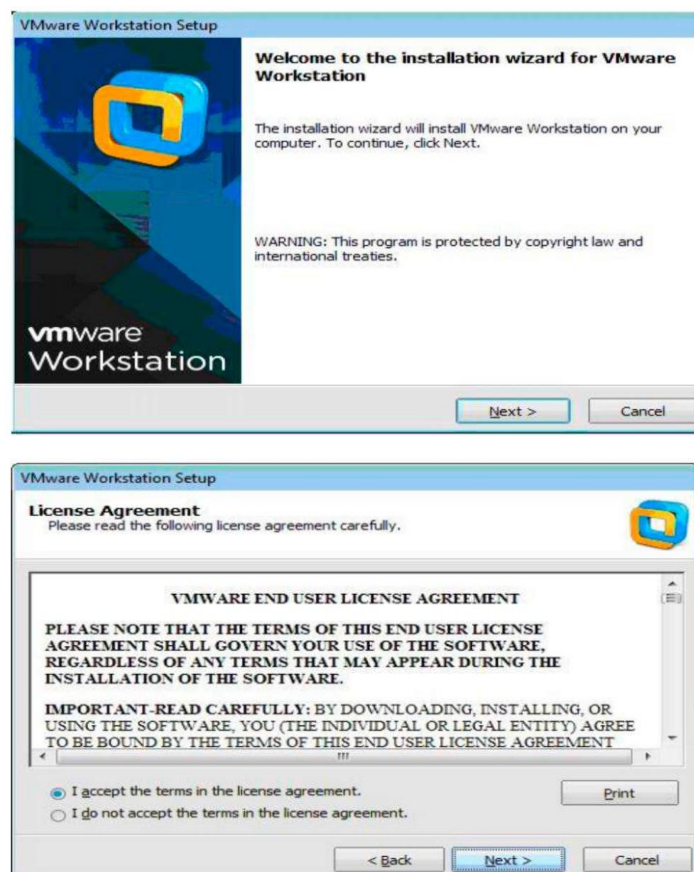


Рисунок 2.1 – Вікно інсталятора та ліцензійна угода VMware Workstation



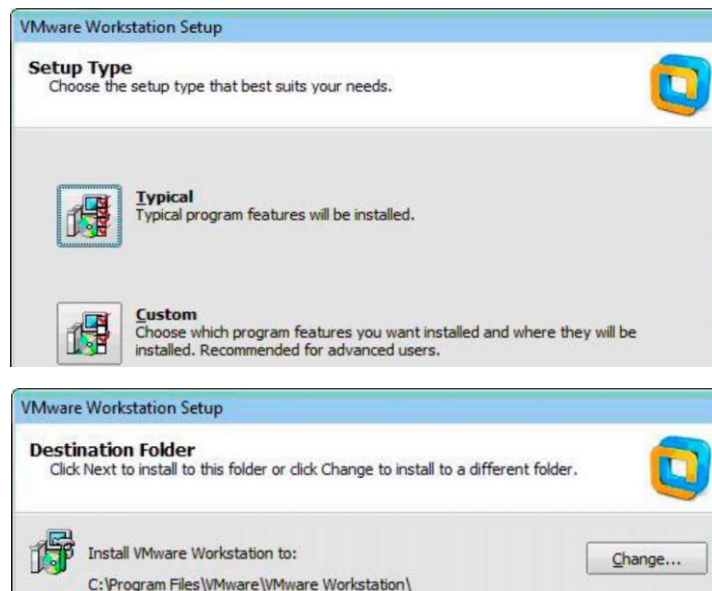


Рисунок 2.2 – Вибір типу налаштування та визначення теки призначення

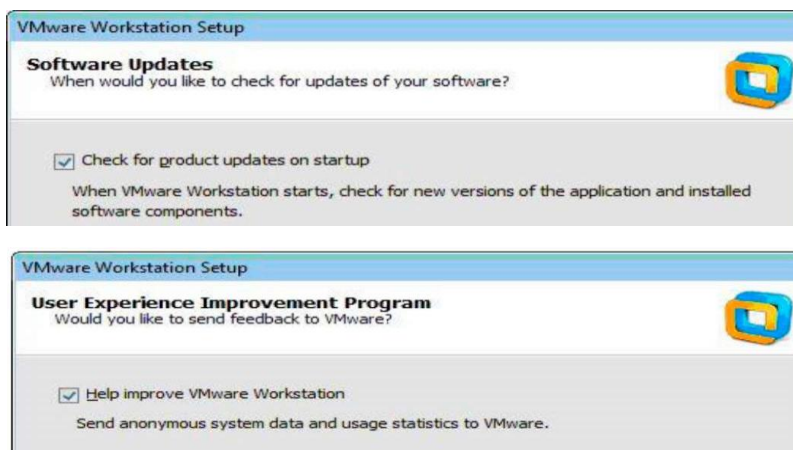


Рисунок 2.3 – Оновлення ПЗ та поліпшення якості програми

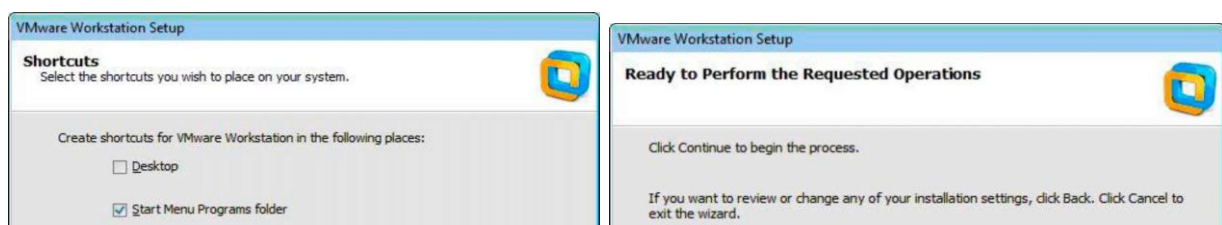


Рисунок 2.4 – Створення ярликів і виконання необхідних операцій

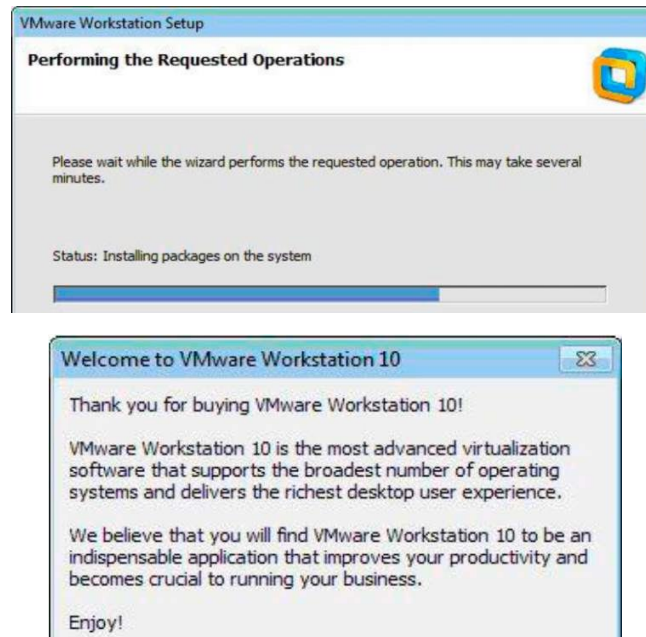


Рисунок 2.5 – Виконання операції та закінчення налаштування

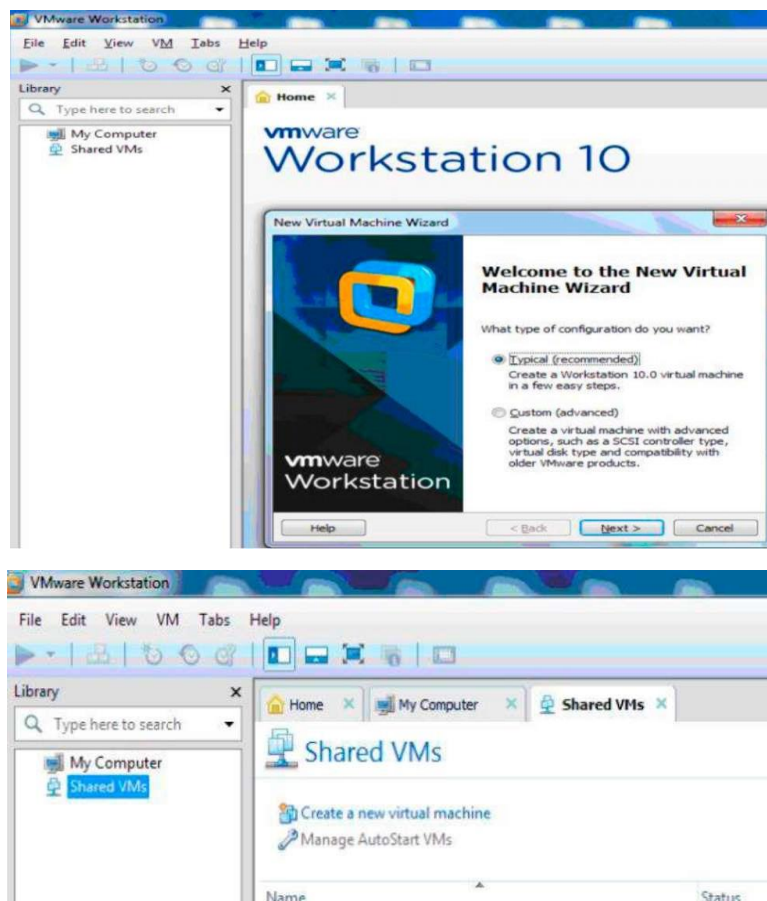


Рисунок 2.6 – Вибір конфігурації VM та Shared VMs

## 2.2.2 Мережеві налаштування

1 Відкрити Пуск → Усі програми → VMWare → Virtual Network Editor і провести конфігурацію віртуальної мережі (рисунок 2.7). У меню File → New → Virtual Machine необхідно налаштувати операційну систему Windows 8 і створити нову віртуальну машину (рисунки 2.7–2.11).

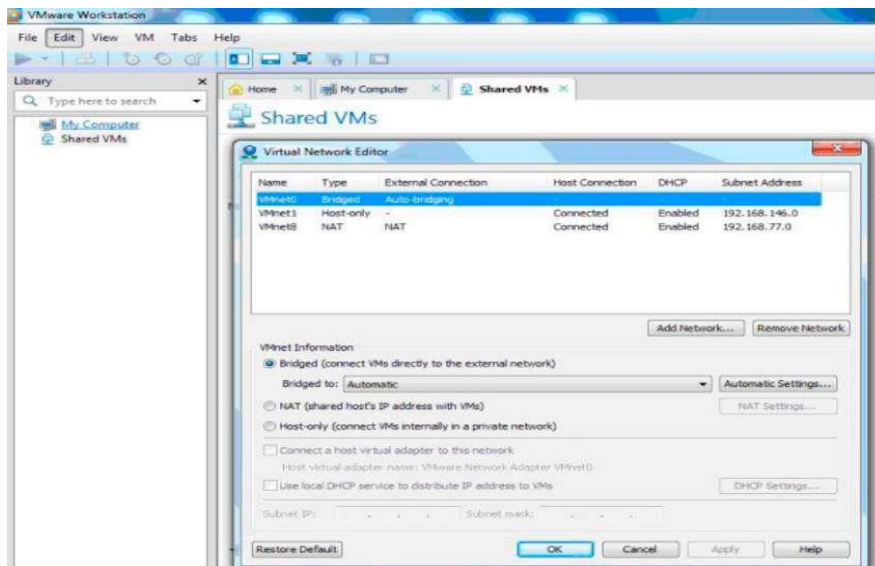


Рисунок 2.7 – Virtual Network Editor

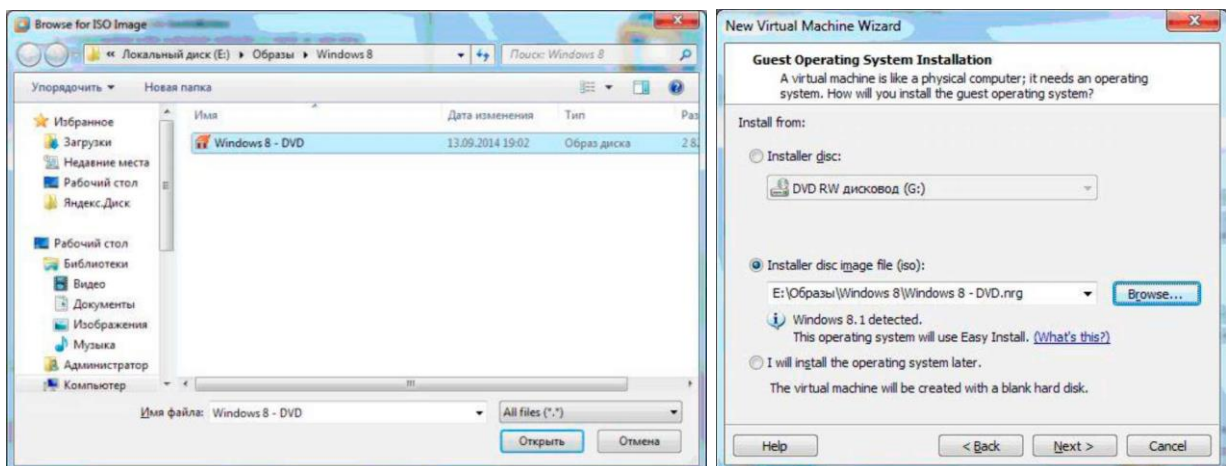


Рисунок 2.8 – Вибір ОС і створення нової VM

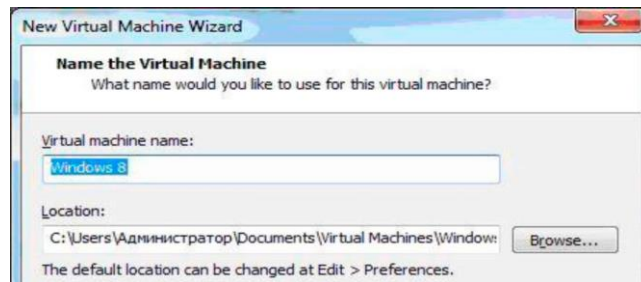
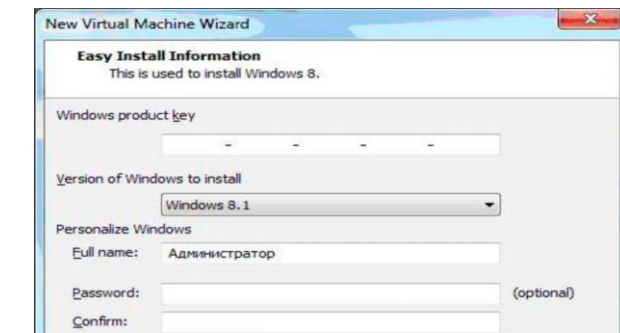


Рисунок 2.9 – Введення ключа продукту та назва ВМ

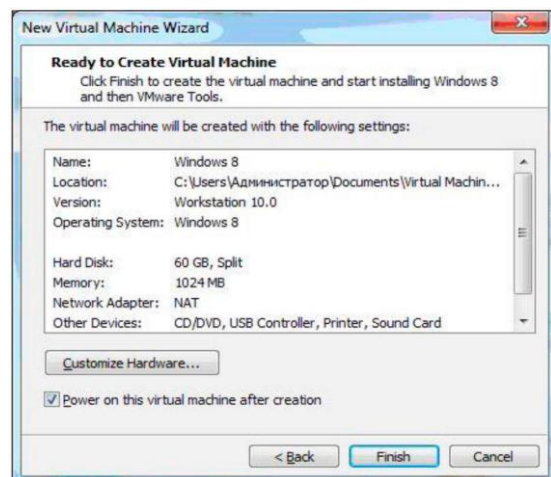


Рисунок 2.10 – Розмір розподіленого диска і створення ВМ

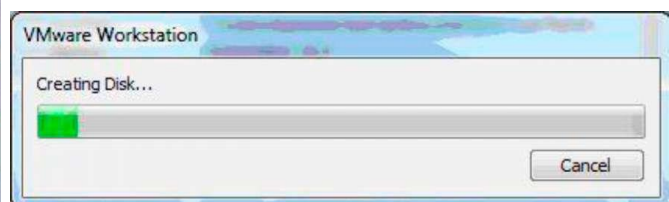
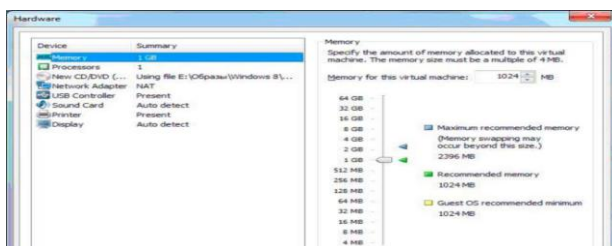


Рисунок 2.11 – Апаратні засоби і створення диска

## 2 Інсталяція на створену VM ОС Windows 8 (рисунок 2.12).

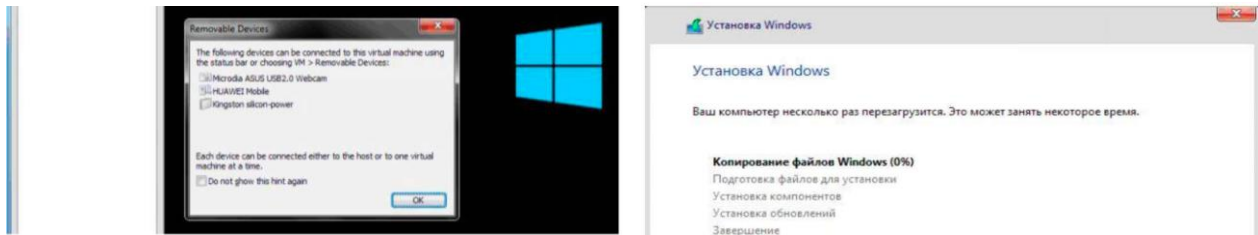


Рисунок 2.12 – Інсталяція Windows S

## 3 Інсталяція та підготовка додатків (рисунок 2.13).

4 Інсталяція особистих параметрів та інстальована операційна система подана на рисунку 2.14.

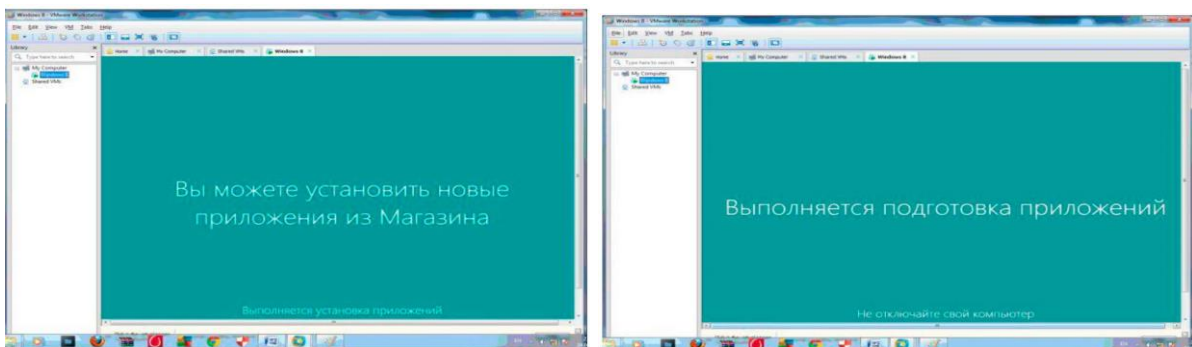


Рисунок 2.13 – Виконання інсталяції та підготовка додатків

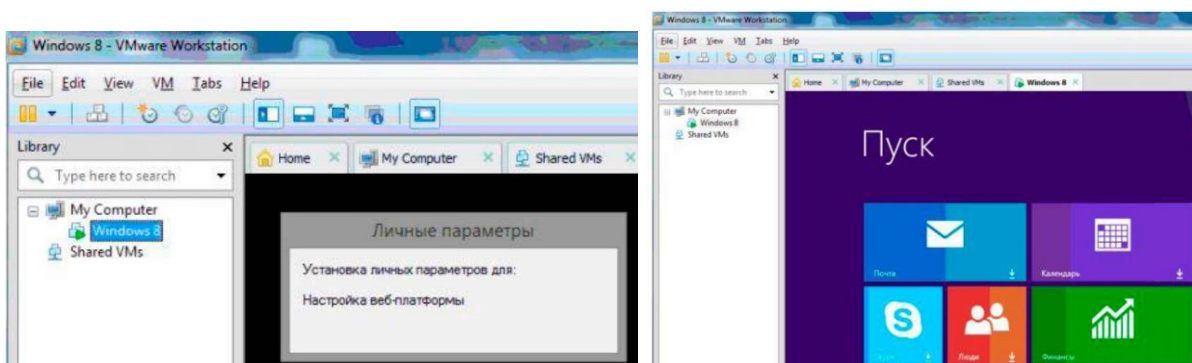


Рисунок 2.14 – Інстальована ОС

### 2.2.3 Установлення інструментів VMWare Workstation

1 Після інсталяції ОС Windows 8 необхідно встановити інструменти VMWare у меню VM → Install VMWare Tools і перезавантажити ОС (рисунок 2.15).

Вікно привітання ОС і запуск ОС наведено на рисунку 2.16.

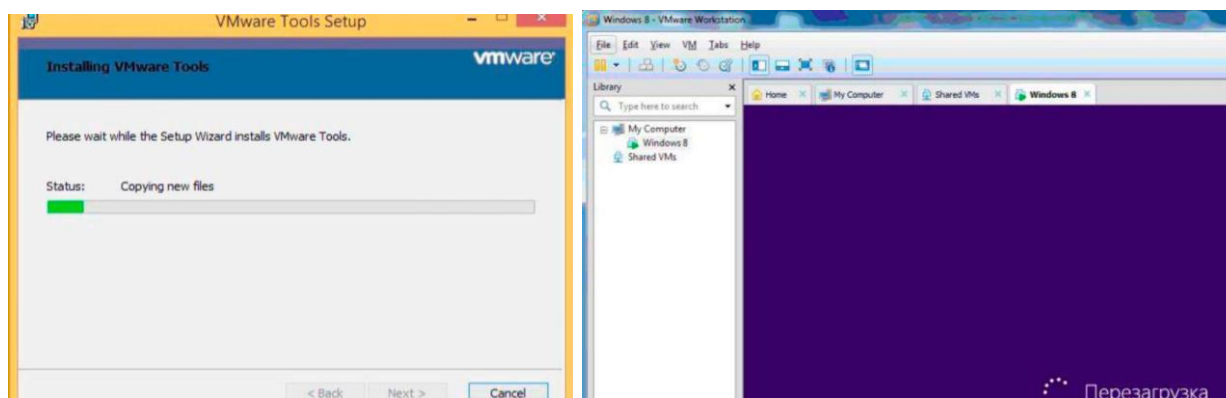


Рисунок 2.15 – Встановлення інструментів VMWare WS і перезавантаження ОС

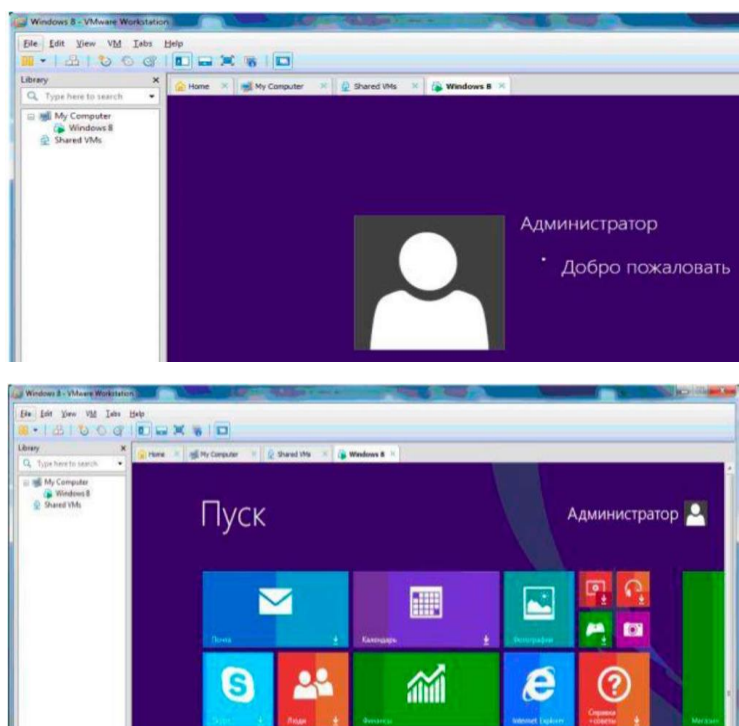


Рисунок 2.16 – Вікно привітання та запуск ОС

2 Створення Snapshot VM. Snapshot VM можна створити, використовуючи VM → Snapshot → Take Snapshot (рисунок 2.17).

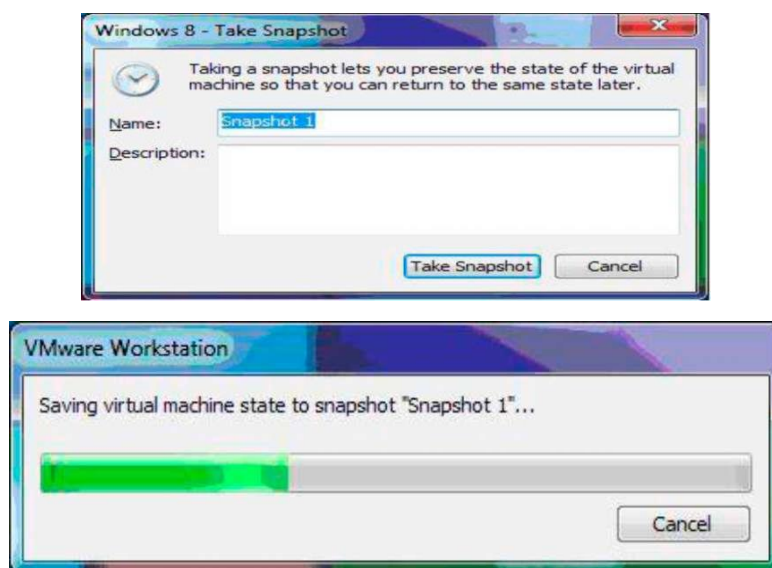


Рисунок 2.17 – Створення Snapshot VM

3 Довільні зміни віртуальної машини. Виконуючи довільні зміни VM, можна скопіювати на робочий стіл декілька ярликів, створити декілька тек (рисунок 2.18). Якщо використати повернення до попереднього Snapshot, то можна відмінити зміни. Повернення до попереднього Snapshot VM можна виконати в меню VM → Snapshot → Revert to Snapshot та обрати попередній Snapshot (рисунок 2.18). Вимкнення та завершення роботи VM показано на рисунку 2.19.

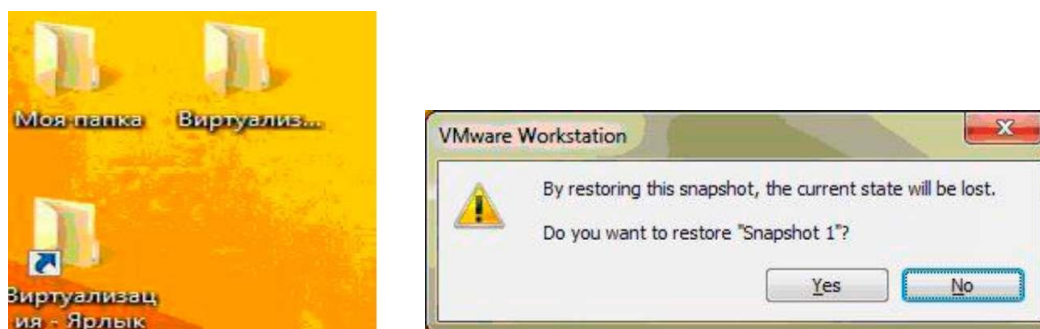


Рисунок 2.18 – Виконання змін VM і повернення до попереднього Snapshot

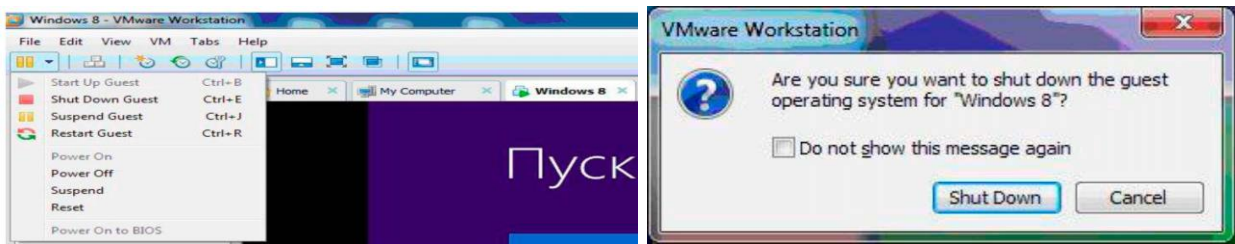


Рисунок 2.19 – Вимкнення та завершення роботи VM

## 2.2.4 Зміни конфігурації VM за допомогою VMWare WS

Зміна конфігурації VM здійснюється в меню VM - Settings. Необхідно збільшити оперативну пам'ять, кількість процесорів, створити додатковий жорсткий диск (рисунки 2.20-2.23).

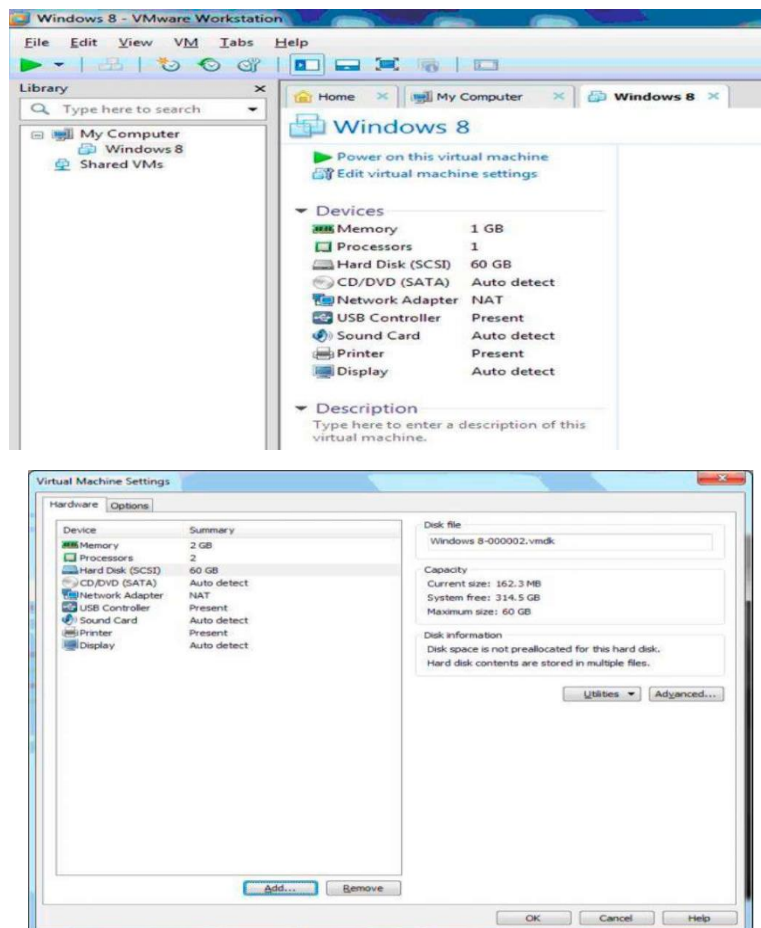


Рисунок 2.20 – Зміна конфігурації VM та збільшення ОП, кількості процесорів



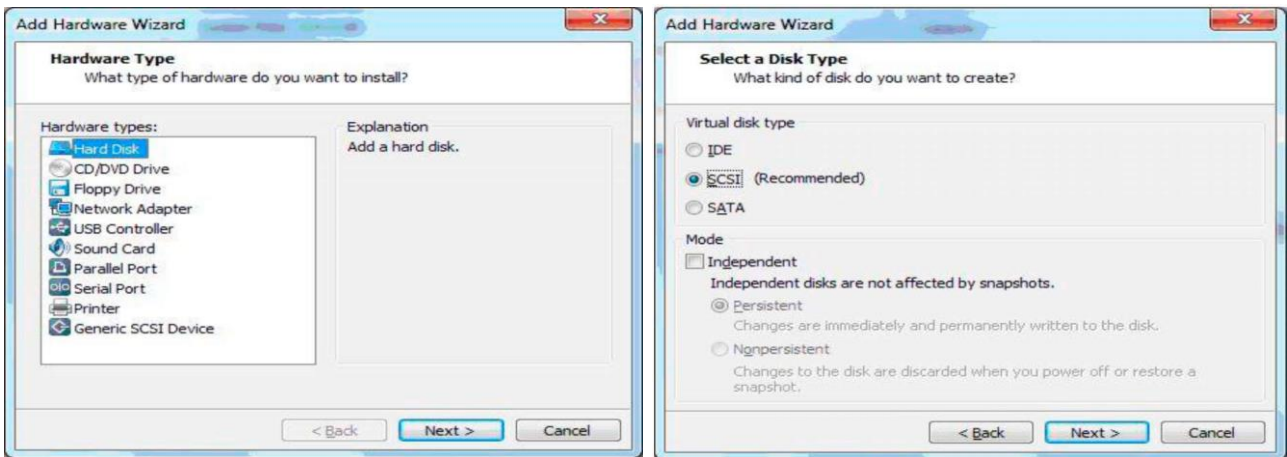


Рисунок 2.21 – Створення додаткового жорсткого диска SCSI

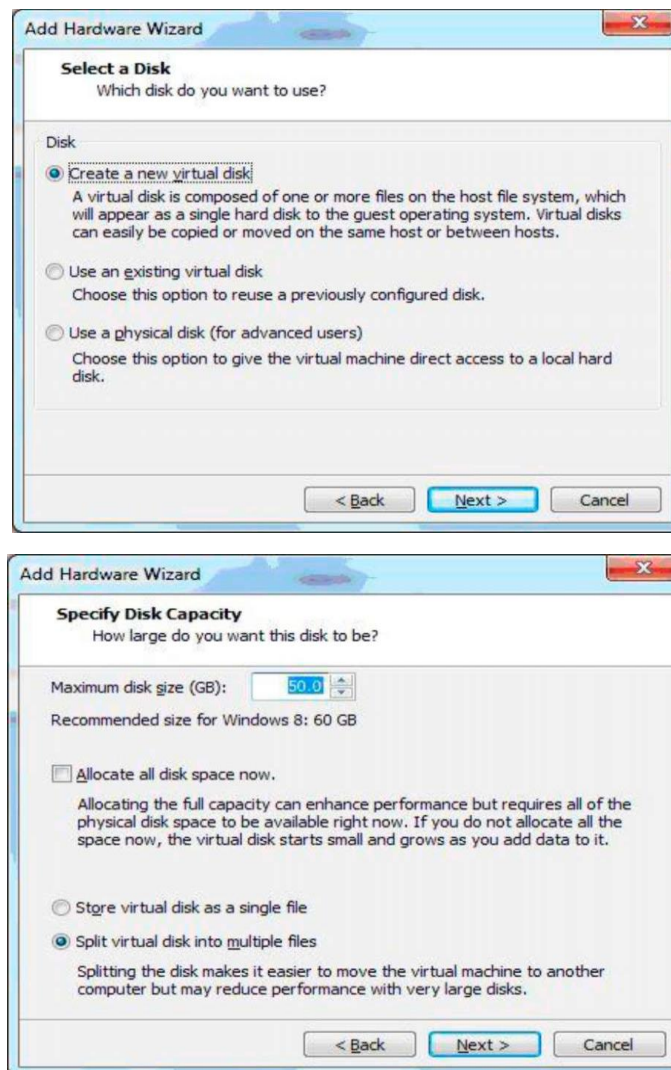


Рисунок 2.22 – Створення та розмір нового віртуального жорсткого диска

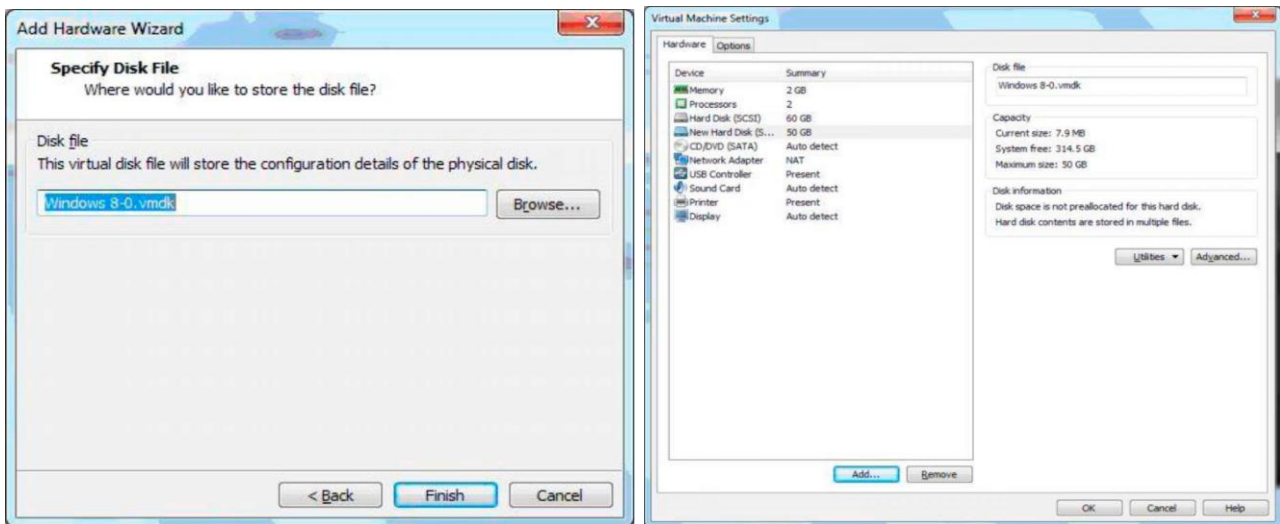


Рисунок 2.23 – Windows S-G.vmdk і створений новий жорсткий диск

## 2.3 Зміст звіту

Після закінчення практичного заняття кожен здобувач вищої освіти повинен скласти звіт, який має містити титульний аркуш, мету заняття, хід роботи з Prnt Scrn, зроблені завдання та висновки про виконану роботу. Звіт оформляє кожен здобувач індивідуально.

## Контрольні запитання

- 1 Класифікація паралельних комп'ютерів і систем.
- 2 Векторно-конверсні комп'ютери.
- 3 Технології віртуалізації. Віртуальні машини.
- 4 Порівняльний аналіз віртуальних машин.
- 5 Ефективне використання обчислювальних ресурсів.
- 6 Зниження витрат на програмне забезпечення.

## **Практичне заняття 3. Створення, вивчення та використання вебдодатка WORD PRESS за допомогою хмарного сервісу MICROSOFT AZURE**

**Мета роботи:** ознайомитися з основами створення вебдодатків, вебдодатком Word Press, навчитися налаштовувати вебдодаток (ВД), здійснювати прості операції з ним, встановлювати гостьову ОС на VM, закріпити навички, отримані на першому та другому практичному занятті та самостійно виконати індивідуальне завдання.

### **3.1 Теоретичні відомості про платформу Windows Azure**

Microsoft Azure – операційна система корпорації Microsoft, призначена для розробників додатків хмарних обчислень і покликана спростити процес створення онлайн-додатків [1–7].

Платформа виготовлена з групи із трьох технологій, що забезпечують спеціалізований набір можливостей для розробників. Крім того, платформу Windows Azure можна використовувати в додатках, що працюють локально на комп'ютерах користувачів, і додатках, які працюють у хмарі [1–7].

Платформа Windows Azure складається з таких компонентів:

- Windows Azure – надає середовище виконання для додатків, заснованих на ОС Windows Server, а також місця для зберігання даних. Система працює на VM за допомогою, аналогічною до 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. Microsoft Azure SQL Database – це хмарний сервіс, що надає можливість зберігання й обробки реляційних даних, а також генерації звітності. На відміну від аналогічних баз даних (БД), БД Azure SQL дають змогу користувачам робити реляційні запити до збережених даних, які можуть бути або структуровані, або частково структуровані, або навіть документів. БД SQL має вилучення даних, пошуку, аналізу даних і синхронізацію даних [1–7].

*Створення додатків.* Для створення додатків Windows Azure Platform можуть бути використані всі технології, які можуть бути запуснені на платформі Windows, зокрема технології .net (посилання технології для Windows Azure хмари), також можна використовувати технології Java, PHP, C/C ++ або Python.

*Основні переваги хмарного рішення Windows Azure:* простота, надійність, безпека, ефективність, гнучкість, легкість.

Windows Azure абстрагує ресурси обладнання за допомогою віртуалізації. Кожен додаток, що розгортається у Windows Azure, працює на одній або декількох ВМ. Розгорнуті додатки працюють так, ніби вони розташовані на виділеному комп'ютері, хоча можуть використовувати такі фізичні ресурси, як місце на диску, мережевий канал введення/виведення або ядра центрального процесора (ЦП), разом з іншими ВМ на тому самому фізичному вузлі.

Дві основні переваги рівня абстракції над фізичним обладнанням – мобільність і масштабованість. Віртуалізація служб дає змогу перенести їх на будь-яку кількість фізичних вузлів у центрі обробки даних. Поєднання технологій віртуалізації, неспеціалізованого обладнання, багатокористувацьких середовищ і агрегування попиту дають Microsoft і нашим клієнтам значну економію. Це дає можливість оптимізувати завантаження центру обробки даних, тобто збільшувати співвідношення

корисної роботи і вартості обладнання і відповідно економити значні кошти. Віртуалізація також дає змогу застосовувати як вертикальне, так і горизонтальне масштабування [1–7].

Вертикальне масштабування означає, що зі зростанням потреби ви можете збільшувати кількість ресурсів, наприклад ядер ЦП і пам'яті, на певній ВМ. Горизонтальне масштабування означає можливість додавати нові екземпляри ВМ, які є копіями існуючих служб. Для всіх цих примірників застосовується балансування навантаження на мережевому рівні, щоб вхідні запити розподілялися між ними.

Платформа Windows Azure містить Windows Azure і SQL Azure. Windows Azure надає обчислювальне середовище на основі Microsoft® Windows Server® для розміщення додатків, постійне сховище структурованих і неструктурованих даних, а також засоби асинхронного обміну повідомленнями. Крім того, Windows Azure передбачає ряд служб, що дають змогу підключати користувачів і локальні додатки до хмарних додатків, здійснювати перевірку достовірності, обмін повідомленнями між службами і управління даними, кешування й інші функції, пов'язані з даними. Windows Azure містить також набір служб управління, які дають змогу контролювати всі ці ресурси за допомогою користувацького вебінтерфейсу (вебпорталу) або програмними засобами. У більшості випадків є програмний інтерфейс на основі REST, за допомогою якого можна визначити те, як працюватимуть служби. Більшість завдань управління, які можуть бути виконані на вебпорталі, також реалізуються за допомогою API. По суті, SQL Azure являє собою SQL Server®, що працює у вигляді хмарної служби. І нарешті, існує великий набір засобів і пакетів засобів для розроблення, тестування та розгортання додатків.

Наприклад, розробляти і тестувати додаток ви можете в емульованому локальному середовищі, створеному за допомогою емуляторів обчислень і сховища. Більшість засобів також інтегровані в такі

середовища розробки, як Microsoft Visual Studio®. Крім того, є засоби управління сторонніх постачальників.

**Служби і компоненти Windows Azure.** Набір служб і компонентів, доступних у Windows Azure і SQL Azure, розрахований на різні конкретні застосування. Після передплати на Windows Azure можна вибрати потрібні можливості, а оплата буде проводитися тільки за використувані засоби. Додавати компоненти в передплату і видаляти їх з неї можна в будь-який час. Механізм виставлення рахунків для кожної служби залежить від типу наданих можливостей [1–7].

**Обчислювальне середовище Windows Azure** складається з платформи для додатків і служб, розміщених в одній або декількох ролях. У Windows Azure можна реалізувати різні типи ролей. Емулятор обчислень Windows Azure (вебролі і робочі ролі). Додаток Windows Azure складається з однієї або декількох розміщених ролей, запущених у центрах обробки даних Azure. Як правило, існує хоча б одна веброль, доступна користувачам програми. Веброль підтримується службами IIS 7.0 і ASP.NET. Додаток може містити додаткові ролі, включаючи робочі, які зазвичай використовуються для виконання фонових задач обробки та підтримки для вебролей. Віртуальна машина (роль VM). Ця роль дає змогу розмістити власний екземпляр ОС Windows Server 2008 R2 Enterprise або Windows Server 2008 R2 Standard у центрі обробки даних Windows Azure.

Управління даними Windows Azure, SQL Azure і пов'язані з ними служби передбачають можливості для зберігання даних і управління ними різними способами. Нижчі розглянуто служби і компоненти управління даними [1–7].

**Сховище Windows Azure.** Забезпечує чотири основних служби для тривалого зберігання даних у хмарі. Ці служби підтримують інтерфейс REST, доступ до якого здійснюється як з додатків, розміщених у Windows

Azure, так і локальних (віддалених) додатків. Нижче перераховані чотири служби зберігання:

- служба таблиць Windows Azure надає механізм зберігання даних у вигляді таблиць і підтримує запити для управління даними;

- служба таблиць Azure – це пропозиція NoSQL, що забезпечує зберігання даних без схеми. Вона призначена насамперед для тих випадків, коли необхідно зберігати великі томи даних, водночас їх потрібно легко витягувати й оновлювати;

- служба великих двійкових об'єктів (BLOB) надає ряд контейнерів, призначених для зберігання текстових і двійкових даних. Передбачені як контейнери блокових великих двійкових об'єктів для потокової передачі даних, так і контейнери сторінкових BLOB-об'єктів для операцій довільного читання і запису;

- служба черг надає механізм для надійного і постійного обміну повідомленнями між екземплярами ролей, наприклад між веброллю і робочою роллю [1–7].

*Диски Windows Azure* надають додаткам механізм для підключення однотоминого VHD-диска з файловою системою NTFS як сторінкового великого двійкового об'єкта, а також передачі і завантаження віртуальних дисків через BLOB. БД SQL Azure – це хмарна служба Database Services з високим ступенем доступності та масштабованості, основана на технології SQL Server, яка підтримує популярну модель реляційних баз даних на основі T-SQL. Її можна використовувати з додатками, розміщеними у Windows Azure, та іншими додатками, що працюють локально або розміщені на інших майданчиках. Синхронізація даних SQL Azure Data Sync – це хмарна служба синхронізації даних, заснована на технологіях Microsoft Sync Framework. Ця служба забезпечує двосторонню синхронізацію даних і дає можливість управляти даними, даючи змогу легко розподіляти їх між кількома базами даних SQL Azure, а також між локальними БД і БД SQL Azure [1–7].

**Служба кешування** надає для додатків розподілений кеш у пам'яті з малою затримкою та високою пропускнуою спроможністю, що не потребує установлення й управління, розмір якого динамічно збільшується або зменшується за необхідності. Може використовуватися для кешування даних додатків, отримання відомостей про стан сеансів ASP.NET, а також кешування сторінок. ASP.NET Windows Azure містить кілька мережевих служб, які можуть бути використані для оптимізації продуктивності, реалізації перевірки автентичності та поліпшення керованості розміщених додатків. Ці служби наведено нижчі.

**Мережа кешуючих серверів (CDN)** дає змогу кешувати загальнодоступні статичні дані для додатків у стратегічних точках, розташованих ближче до кінцевих користувачів (з погляду доставки по мережі).

**Служба підключення до віртуальної мережі** забезпечує налаштування ролей додатка, що працює у Windows Azure і на комп'ютерах локальної мережі, що дає змогу уявити, ніби ці ролі розташовані в тій самій мережі. При цьому програмний агент запускається та використовується на локальному комп'ютері, який встановлює захищене з'єднання IPsec з ролями Windows Azure у хмарі, а також забезпечує можливість адміністрування, відстеження, налагодження та управління ролями безпосередньо.

**Диспетчер трафіку віртуальних мереж** дає змогу налаштувати перенаправлення запитів і балансування навантаження на основі трьох різних методів. Як правило, диспетчер трафіка використовується для підвищення продуктивності методом, коли запити перенаправляються до примірника додатка в тому центрі обробки даних, який розміщений найближче до користувача. Існують і інші методи балансування: відпрацювання відмови, циклічний перебір та управління доступом (ACS). Ця служба побудована на основі стандартів, призначена для управління доступом і дозволами, вона використовує безліч постачальників посвідчень (IdP), які можуть виконувати перевірку автентичності користувачів.



**Служба управління доступом (ACS)** діє як служба токенів безпеки (STS) і допомагає використовувати переваги методів федеративної перевірки автентичності, коли справжність користувача перевіряється не в тій області або домені, де розміщений додаток. Як приклад можна навести управління доступом користувачів на основі дозволів, які перевіряються постачальником послуг, наприклад Windows Live® ID або Google.

**Шина обслуговування.** Ця служба забезпечує безпечний обмін повідомленнями і потоками даних у розподілених і гібридних додатках, наприклад для зв'язку розміщених додатків Windows Azure і локальних додатків, виключаючи необхідність реалізації складної інфраструктури брандмауерів і системи безпеки. Вона може використовувати ряд протоколів і шаблонів взаємодії та передачі повідомлень для гарантованої і надійної доставки повідомлень, а також масштабується для різних навантажень та інтегрується з локальними об'єктами BizTalk Server [1–7].

**Управління базами даних SQL Azure.** Додатки отримують доступ до баз даних SQL Azure так само, як і локально встановлених екземплярів SQL Server, за допомогою керованих класів доступу до даних ADO.NET, DAAB, OData, власних класів ODBC, PHP, Ruby або технологій доступу до даних JDBC. Управління БД SQL Azure здійснюється через вебпортал, у середовищі SQL Server Management Studio, за допомогою засобів для роботи з базами даних Visual Studio 2010 та інших інструментів для виконання таких завдань, як переміщення і перенесення даних. Доступні й інші засоби командного рядка для розгортання та адміністрування баз даних. Крім того, доступний диспетчер баз даних, який полегшує роботу з базами даних SQL Azure [1–7].

**Оновлення додатків для Windows Azure.** Після розгортання програми для Windows Azure його необхідно оновлювати при зміні служб ролей у відповідь на нові вимоги та покращення в коді або для виправлення помилок. Ви можете просто виконати повторне розгортання

служби. Для цього потрібно призупинити і видалити її, потім виконати розгортання нової версії. Однак можна уникнути простою додатків завдяки поетапному розгортанню (завантаженню нового пакета і заміни існуючої робочої версії цим пакетом) або оновленню на місці.

Адміністратори управляють окремими розміщеними службами для підписки Windows Azure. Підписка Windows Azure може включати один або декілька подібних елементів:

- розміщення служби, що складаються з розміщених ролей і примірників у кожній ролі; ролі й примірники можуть бути зупинені в робочому режимі або режимі проміжного зберігання;

- облікові записи зберігання, що складаються з примірників служб таблиць, великих двійкових об'єктів і черг.

**Примірники CDN.** CMS (Content Management System) перекладається як система управління змістом. Основні функції системи управління CMS: зручний інструмент для додавання нової інформації до сайту, редагування та видалення існуючої інформації, управління функціоналом сайту.

Для створення сайту така система не обов'язкова, але для кожного сайту важливе його своєчасне оновлення і розширення. Крім того, має бути можливість створення контенту сайту без навичок програмування. Тому сьогодні майже всі сайти редагуються за допомогою CMS. Існує багато різних CMS. Їх розрізняють за вільністю використання (безкоштовні, платні та внутрішні), призначенням (універсальні, призначені для блогів, інтернет-магазинів тощо), способом створення сторінок (попереднє створення, «на льоту» або змішане). Приклади найбільш відомих CMS: Drupal, Joomla!, Wordpress, Magenta. Wordpress – це CMS, призначена для блогів. Вона вважається однією з найпростіших для користувача, що пояснює її популярність. На основі Wordpress можна створити будь-який сайт, але більшість її функцій

призначені саме для блогів. Однією з переваг є легке налаштування за допомогою численних тем, плагінів і віджетів [1–7].

## 3.2 Порядок виконання роботи

### 3.2.1 Створення вебдодатків служби додатків Azure

1 Виконати вхід на порталі Azure, як наведено на рисунку 3.1.

2 Відкрити Azure Marketplace, натиснувши значок Marketplace, як наведено на рисунку 3.2. Крім того, можна натиснути значок Створити в правому верхньому куті приладової панелі і вибрати Marketplace у нижній частині списку.

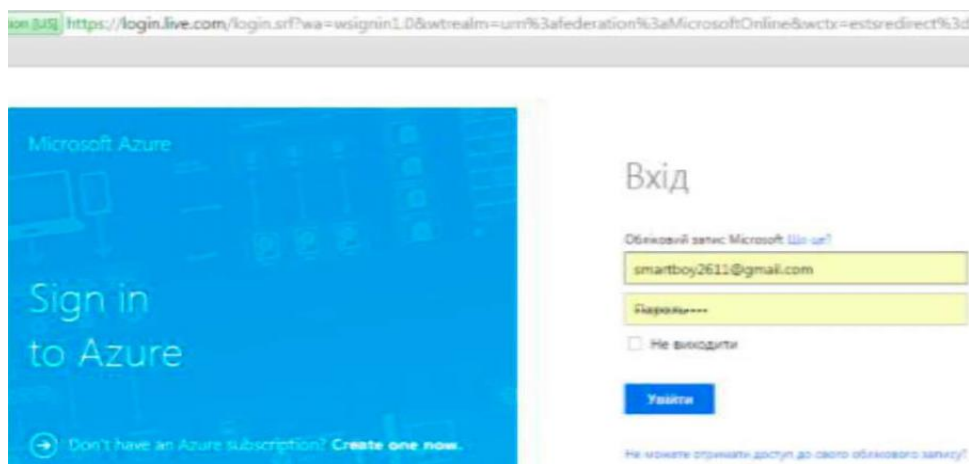


Рисунок 3.1 – Вхід на портал Azure

3 Вибрати Інтернет + мобільні пристрої. Знайдіть WordPress і натисніть значок WordPress, як зазначено на рисунку 3.3.

4 Прочитавши опис додатку WordPress, натиснути значок Створити. Натиснути на значок вебдодатки і вказати необхідні значення для налаштування вебдодатків, як наведено на рисунку 3.4.

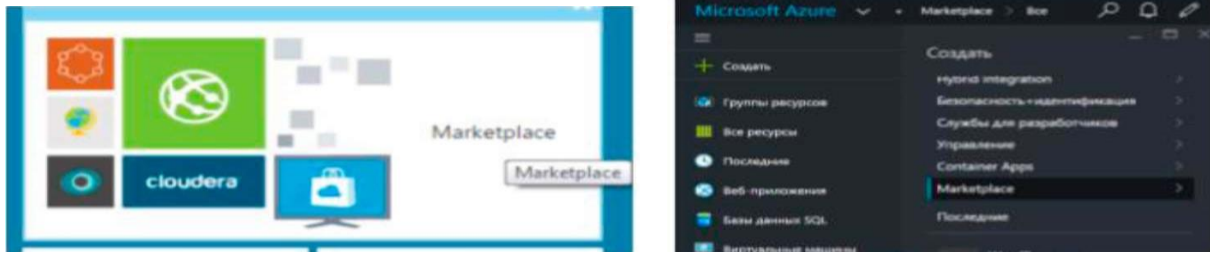


Рисунок 3.2 – Azure Marketplace

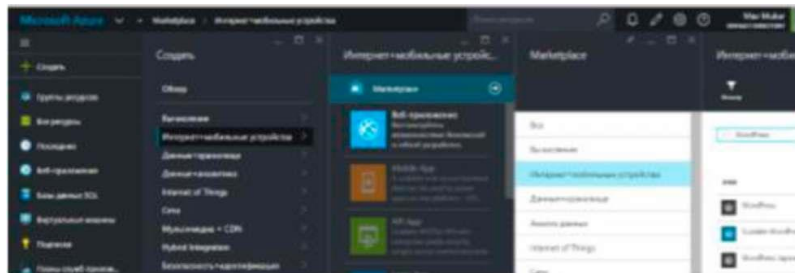


Рисунок 3.3 – Інтернет + мобільні пристрої



Рисунок 3.4 – Налаштування вебдодатків

5 Натиснути на елемент БД і вказати в колонці Нова БД MySQL необхідні значення для налаштування БД MySQL [8–14]:

- ввести нове ім'я або використати стандартне ім'я;
- у полі Тип БД залишити стандартне значення;
- вибрати цінову категорію. Для цього для нас прийнятним є Меркурій (безкоштовний з мінімум дозволених підключень і місця на диску).

6 У колонці Нова БД MySQL натиснути ОК.

7 У колонці WordPress прийняти усі умови, що пропонуються, і натиснути кнопку Створити, як наведено на рисунку 3.5.

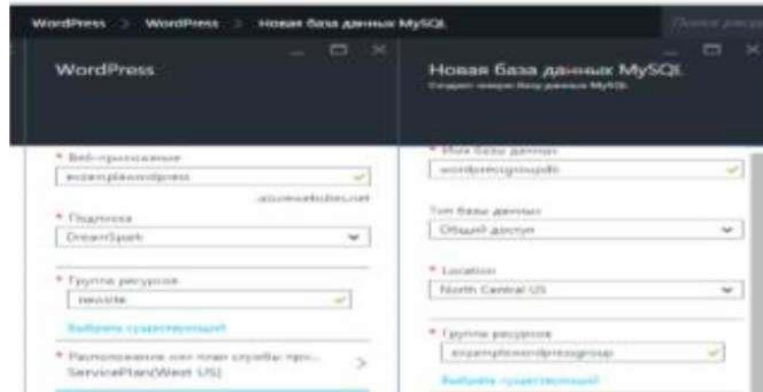


Рисунок 3.5 – Створення нової БД

8 У службі додатків Azure створити вебдодаток (зазвичай це займає менше хвилини). Можна відстежувати хід виконання, натиснувши значок дзвіночка у верхній частині сторінки порталу, як наведено на рисунку 3.6.



Рисунок 3.6 – Створення вебдодатка

### 3.2.2 Запуск веб додатка WordPress і управління ним

1 Після завершення створеного вебдодатка перейти на порталі до групи ресурсів, у якій було створено програму, і ви побачите вебдодаток і БД, як наведено на рисунку 3.7:

- у колонці Група ресурсів натиснути на рядок вебдодатки [8–14];
- натиснути на вебдодаток для перегляду відомостей про програму.

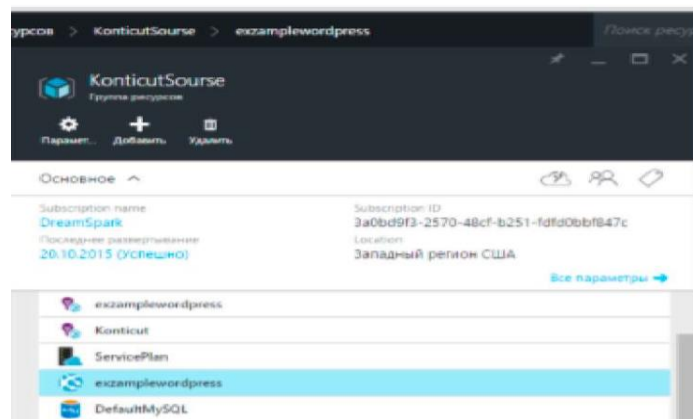


Рисунок 3.7 – Вебдодаток і БД

### 3.2.3 Використання хмарного сервісу MICROSOFT AZURE

1 Виконати вхід на порталі Azure.

2 Відкрити Azure Marketplace, натиснувши значок Marketplace.

Крім того, можна натиснути значок Створити в правому верхньому куті приладової панелі і вибрати Marketplace у нижній частині списку.

3 Вибрати Інтернет + мобільні пристрої. Знайти WordPress і натиснути значок WordPress.

4 Прочитавши опис додатка WordPress, натиснути Створити.

5 Натиснути вебдодатки і вказати необхідні значення для налаштування.

6 Натиснути на елемент БД і вказати у колонці Нова БД MySQL необхідні значення для налаштування БД MySQL:

– ввести нове ім'я або використати стандартне ім'я;

– у полі Тип БД залишити стандартне значення;

– вибрати цінову категорію – для нас є прийнятним «Меркурій» (безкоштовний з мінімумом дозволених підключень і місця на диску).

7 У колонці Нова БД MySQL натиснути ОК.

8 У колонці WordPress прийняти умови, що пропонуються, і натиснути кнопку Створити.

Служба додатків Azure створить вебдодаток. Можна відстежувати хід виконання, натиснувши значок дзвіночка у верхній частині сторінки порталу.

**Запуск вебдодатка WordPress і управління ним.** Після завершення створеного вебдодатка перейти на порталі до Grid-ресурсів, у якій було створено програму, і проаналізувати вебдодаток та БД:

- у колонці Група ресурсів натиснути на рядок вебдодатка [8–14];
- натиснути на вебдодаток для перегляду відомостей про програму.

### **3.3 Зміст звіту**

Після закінчення практичного заняття кожен здобувач вищої освіти повинен скласти звіт, який має містити титульний аркуш, мету заняття, хід роботи і Prnt Scrn (прінскріни) зроблених завдань і висновки про виконану роботу.

Звіт оформляє кожен здобувач вищої освіти індивідуально.

### **Контрольні запитання**

- 1 Підвищення гнучкості і швидкості реагування системи.
- 2 Чи можуть несумісні додатки працювати на одному комп'ютері?
- 3 Віртуалізація застосувань.
- 4 Віртуалізація уявлень (робочих місць).

## **Практичне заняття 4. Архітектура і процес автоматизованого розгортання віртуальних машин. Створення автошаблону віртуальних машин**

**Мета роботи:** практичне засвоєння технології віртуалізації та набуття навичок створення шаблонів VM.

**Основні навички:** спосіб автоматичного розгортання VM, що дає змогу швидко запускати та активувати набори автоматично налаштованих VM.

**Апаратне і програмне забезпечення:** настільний або портативний комп'ютер, який підтримує віртуалізацію.

**Завдання:** інсталювання диспетчера налаштування VM під Linux®, Windows®, створення шаблону VM, який автоматично налаштовується.

### **4.1 Теоретичні відомості про автоматичне розгортання VM**

Автоматичний спосіб розгортання VM складається з двох основних частин: розгортання менеджера VM (менеджер розгортання віртуальної машини, VDM) і менеджера конфігурації VM (менеджер конфігурації VM, VCM). VDM обробляє запити користувача для розгортання VM (VM), а саме клонування VM-образів, налаштування устаткування VM та реєстрація VM у гіпервізорі. VCM встановлюється на образ VM, який використовується як шаблон для VM, після завантаження системи він автоматично налаштовує VM [1–7].

### **4.2 Порядок виконання роботи**

#### **4.2.1 Автоматичний спосіб розгортання віртуальних машин**

Архітектура та процес автоматичного розгортання VM показано на рисунку 4.1.



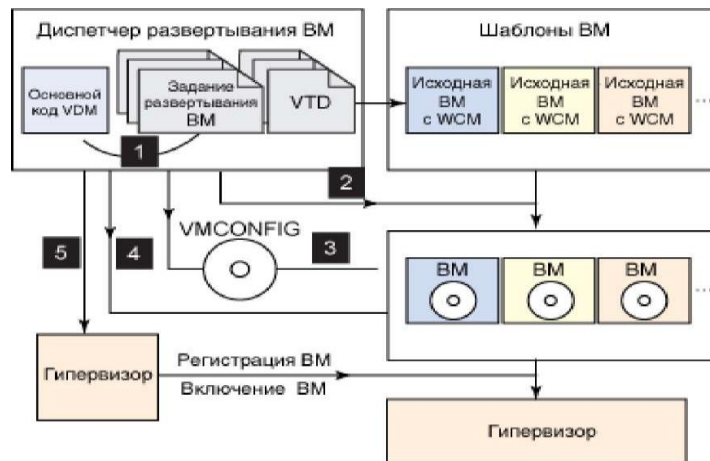


Рисунок 4.1 – Архітектура та процес автоматичного розгортання VM

Архітектура та процес автоматичного розгортання VM складається з трьох частин:

- основний код VDM, VDM ядра;
- завдання розгортання VM – файл опису окремого завдання розгортання VM, різновид файлу конфігурації користувача;
- визначення шаблону VM (VM шаблон визначень, VTD) – файли, які містять інформацію VM шаблонів, а саме розташування й опис шаблону.

VM шаблон – це VM, яка використовується як джерело для клонування VM [8–14].

Основна частина процесу розгортання VM складається з таких етапів:

- 1 VDM зчитує інформацію з VM завдання розгортання.
- 2 VDM шукає відповідний шаблон VM у каталозі, який містить файли і шаблон клонів VTD VM розташування, визначене в завданні розгортання файлу.
- 3 Після клонування оригінал VM VDM збирає всі дані конфігурації, пакує їх у файл ISO і потім копіює його в каталог з копією до VM.

4 VDM змінює параметри обладнання клону VM на основі налаштування, зазначеного в завданні розгортання VM. Він також створює компакт-диск для монтування ISO-файлу, який був створений у кроці 3.

5 Остання стадія – розгортання утиліти гіпервізора для реєстрації VM і її подальшого запуску.

Пункт 5 не є обов'язковим, користувач може пропустити ці кроки в завданні розгортання VM. Як гіпервізор може виступати VMware Server, Xen та ін. [8–14].

VCM встановлюється на VM як шаблон. Після перемикання в системі він запускається автоматично, виконує пошук даних конфігурації CD, і потім починається запуск необхідних додатків для виконання всіх попередніх дій.

#### 4.2.2 Створення автошаблону віртуальних машин

Щоб створити шаблон VM, який автоматично налаштовується, треба після початкового налаштування VM інстальювати диспетчер налаштування VM (Virtual Machine Configuration Manager, VCM), як показано на рисунку 4.2.

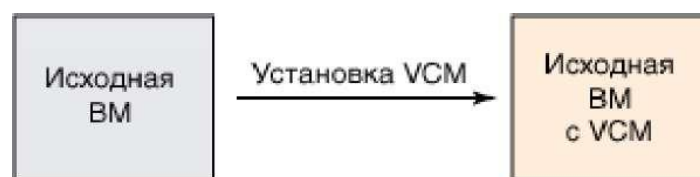


Рисунок 4.2 – Диспетчер налаштування VM

Основною функцією в VCM є знаходження конфігураційного CD з лейблом VMCONIFG і запуск з нього необхідних додатків після ввімкнення системи. Оскільки початкова VM працює під Linux®, Windows® або іншою операційною системою, то VCM для різних операційних систем теж розрізняється [8–14].

**1 Менеджер налаштування ВМ для Linux.** Після встановлення менеджера налаштування ВМ для Linux він є зареєстрованим як служба з Linux, працює під час першого завантаження. Система виконує бажані налаштування і потім, після завершення конфігурації, виконує дереєстрацію та видаляє себе. Щоб установити VCM для Linux, треба скопіювати пакет VCM до оригінальної ВМ, розпакувати його в будь-яку директорію і запустити скрипт інсталяції.

У лістингу 4.1 наведено уривок коду `install.sh`.

Оскільки всі необхідні додатки і дані зберігаються на CD-конфігурації, VCM шукає цей диск, монтує його в `/media /VMCONFIG` і потім виконує необхідну конфігурацію.

Лістинг 4.2 показує, як VCM для Linux шукає конфігурацію CD після завантаження системи.

Лістинг 4.1. Уривок коду скрипту інсталювання VCM `install.sh`.

```
@echo off
1
2 ...
3   VCM_HOME=$(cd $(dirname $0); pwd)
4   start='getStartNum'
5   stop='getStopNum'
6   sed -e "s!^VCM_HOME=.!*VCM_HOME=$VCM_HOME!\
7s!^# chkconfig: 35!# chkconfig: 35 $start
$stop!"\
8 "$VCM_HOME/vmconfigmgr" > "/etc/init.d/vmconfigmgr"
9   chmod +x "/etc/init.d/vmconfigmgr"
10  echo "Register service vmconfigmgr"
    chkconfig --add vmconfigmgr
11  if [ $? -eq 0 ]; then
```

```

11 echo "Install Complete Successfully!"
12  else
13 echo "Install Failed."
14  Fi
15
-   @echo on

```

Лістинг 4.2. Приклад коду, який виконує пошук і монтаж конфігурації CD.

```

@echo off
1
2 LABEL="VMCONFIG"
3 VM_CFG_DIR=/media/$LABEL
4 for device in `dmesg | grep "A.*:.*CD-ROM" |
awk -F':' '{print $1}'
do
5 volumeName=Nvolname "/dev/$device" | awk
'{{print $1}},
6 PrintString "CD-ROM Drive: $device | Label:
$volumeName"
7 if [ "$volumeName" == "$LABEL" ];then
PrintString "VM Configuration CD-ROM is:
/dev/$device"
8 # Mount the CD-ROM
9 mkdir -p "$VM_CFG_DIR"
10 mount -t iso9660 -o ro,nosuid,nodev,utf8,uid=0
"/dev/$device"
11 "$VM_CFG_DIR"

```

```
12 break
13 done
14
15
@echo on
```

## 2 Менеджер конфігурації віртуальної машини для Windows.

Після встановлення менеджера конфігурації ВМ для Windows створює заплановане завдання, яке запускається один раз після завантаження комп'ютера. Як і VCM для Linux, Windows VCM також видаляє заплановане завдання після того, як воно буде завершено [8–14].

Для інсталяції VCM для Windows необхідно скопіювати архів з його дистрибутивом до шаблону ВМ Windows, розпакувати його в будь-який каталог і запустити скрипт Install.bat. Лістинг 4.3 показує приклад Install.bat.

Як і Linux VCM, Windows VCM необхідно знати букву пристрою з компакт-дисків конфігурації. Windows VCM використовує для цього скрипт WMI, як показано і лістингу 4.4.

### Лістинг 4.3. Зразок коду скрипту інсталяції VCM Windows.

```
@echo off
1
2 set VCM_HOME=%~dp0
3 REM Create scheduled task to run VM
Configuration Manager when system
4 boots
5 schtasks /create /tn "VMCONFIG" /tr
%VCM_HOME%vmconfigmgr.bat /sc
6 onstart /ru "System"
7
@echo on
```

Лістинг 4.4. Зразок VB коду скрипту, який використовується для визначення букви конфігурації CD.

```
@echo off
1
Function getDriveLetter(label)
2 Const wbemFlagReturnImmediately = &h10
3 Const wbemFlagForwardOnly = &h2 0
4 strComputer="."
5 driveLetter=""
6 Set objWMIService = GetObject("winmgmts:\"" &
strComputer "\root\CIMV2")
7 Set colItems = objWMIService.ExecQuery("SELECT
* FROM
8 Win32_CDROMDrive", "WQL", _
9
wbemFlagReturnImmediately
+
10
wbemFlagForwardOnly)
11 For Each objItem In colItems
WScript.Echo "Drive: " & objItem.Drive & "
VolumeName: "
& objItem.VolumeName & " Status: "
objItem.Status
14 If StrComp(objItem.VolumeName,
label) = 0 Then
15 driveLetter=objItem.Drive
16 WScript.Echo "driveLetter:
" & driveLetter
17 Exit For
```

```
18             End If
19         Next
20         getDriveLetter=driveLetter
21     End Function
22 =getDriveLetter("VMCONFIG")
@echo on
```

Після інсталяції VCM на оригінальну ВМ він стає автоматично налаштовуваним шаблоном ВМ [8–14].

### **4.3 Зміст звіту**

Після закінчення практичного заняття кожен здобувач вищої освіти повинен скласти звіт, який має містити титульний аркуш, мету заняття, хід роботи зі скринами (Print Screen) зроблених завдань і висновки про виконану роботу. Звіт оформляє кожен здобувач вищої освіти індивідуально.

&

### **Контрольні запитання**

- 1 Віртуалізація серверів.
- 2 Паравіртуалізація.
- 3 Віртуалізація на рівні ядра ОС.
- 4 Технології віртуалізації.
- 5 Переваги хмарних обчислень.
- 6 Недоліки та проблеми хмарних обчислень.

## **Практичне заняття 5. Створення файлів визначення та завдань. Розгортання віртуальної машини**

**Мета:** практичне освоєння технології віртуалізації та набуття навичок створення файлів визначення і завдань розгортання VM.

**Апаратне і програмне забезпечення:** настільний або портативний комп'ютер, який підтримує віртуалізацію.

**Завдання:** створення та налаштування основних і мережевих параметрів VM у файлі завдання розгортання параметрів індивідуального користувача; параметрів реєстрації та запуску віртуальної машини.

### **5.1 Теоретичні відомості**

Шаблони VM зберігаються окремо, тому для VDM потрібен механізм для їхнього виявлення. Для цього використовується файл визначення, який описує властивості VM-шаблону. Один файл визначення відповідає одному шаблону. Усі файли визначення зберігаються в каталозі vtfs машини VDM. Файли визначення мають імена типу OSName.vtd (де OSName використовується для диференціювання шаблонів VM). Наприклад, щоб указати на шаблон VM SLES 10 SP2, можна використовувати назву файлу визначення SLES10SP2.vtd ім'я [1–7].

### **5.2 Порядок виконання роботи**

#### **5.2.1 Створення файлів визначення віртуальної машини**

Лістинг 5.1 показує приклад файлу визначення шаблону VM SLES10 SP2.



## Лістинг 5.1. Приклад файлу визначення шаблону VM SLES10 SP2.

```
@echo off
1 # Файл определение шаблона VM
2 # Тип ОС шаблона VM, допустимые значения
Windows, Linux.
3 OSType=Linux
4 # Описание шаблона VM.
5 Description=SuSE Linux Enterprise Server 10 SP2
# Каталог, в котором расположен шаблон VM.
6 SRCVMDIR=/local/vmware/SuSE Linux Enterprise
Server 10 SP2
7
@echo on
```

### 5.2.2 Створення завдань розгортання VM

Параметри VM вказані у файлі завдання розгортання VM. Усі команди конфігурації для розгортання вказуються в одному файлі, поділяються на декілька категорій: основні параметри VM; налаштування мережі; параметри індивідуального користувача; параметри реєстрації та запуску VM [8–14].

У лістингах 5.2-5.5 відповідні параметри наведені для завдання розгортання *sles10sp2vm.task*.

Ці основні параметри визначають тип ОС, розмір оперативної пам'яті, диска і розміщення створюваної VM. Крім того, можна вказати, чи слід перезаписати VM, якщо вона вже існує у вказаному каталозі.

Під час розгортання інформація про налаштування мережі зібрана за допомогою VDM і збережена в образі ISO CD конфігурації. Відповідно вона доступна при запуску VM [8–14].

Уся інформація про параметри користувача також буде зібрана і зберігатиметься в образі ISO CD конфігурації.

Ці параметри використовуються для реєстрації або увімкнення VM після розгортання. Наприклад, для VMware Server 2.0 ви повинні вказати URL-адресу сервера, ім'я користувача і пароль [8–14].

#### Лістинг 5.2. Основні параметри VM у файлі завдання розгортання.

```
@echo off
1 # Описание задания
2 TaskDescription=SLES 10 VM used for BVT
3 # Название ОС создаваемой VM, в каталоге vtds
должен присутствовать файл с
4 # соответствующим именем.
5 OSName=SLE S10SP2
6 # Размер памяти в МБ, выделенный для VM
7 MemorySize=1300
8 # Размер в ГБ первого жёсткого диска, если
указан, если не указан или
9 # меньше, чем размер диска шаблона VM,
создается диск такого же размера, как
10 # в шаблоне VM.
11 DiskSize=10
12 # Отображаемое название VM.
13 DisplayName=SLES10SP2-143-50
14 # Каталог, в котором должна быть создана VM.
15 Destination=/ vmware/SLE S10SP2-143-50
16 # Нужно ли перезаписывать существующую VM.
17 OverwriteOnExist=Yes
@echo on
```

### Лістинг 5.3. Мережеві параметри ВМ у файлі завдання розгортання.

```
@echo off
1 # Определяет, нужно ли настраивать сетевые
параметры при первом запуске ВМ.
2 # Yes - настраивать, No - не настраивать.
3 # Если указано значение No, все свойства,
начиная с ConfigNet, будут проигнорированы.
ConfigNet=yes
4 # Тип сетевого подключения виртуального
сетевого адаптера eth0, допустимые значения
5 # Bridged (мост), NAT и HostOnly. Если параметр
не указан, по умолчанию используется
6 # значение Bridged.
7 ConfigNet.ConnectType=NAT
8 # MAC-адрес сетевого адаптера, допустимые
значения:
9 # 1) auto автоматическая генерация MAC.
10 # 2) xx:xx:xx:xx:xx:xx пользовательский MAC.
(VMware соответствует диапазон
11 # с 00:50:56:00:00:00 по 00:50:56:3F:FF:FF)
ConfigNet.MacAddress=00:50:56:3A:01:02
12 # Метод настройки сетевых параметров, dhcp или
static.
13 ConfigNet.Mode=static
14 # IP-адрес используемый ВМ.
15 ConfigNet.IPAddress=192.168.143.50
16 ConfigNet.Netmask=255.255.255.0
17 ConfigNet.Gateway= 192.168.143.2
18 ConfigNet.Hostname=sles10vm-143-50
```

```
19 ConfigNet.Domain=ibm.com
20 ConfigNet.PrimaryDNS=9.181.32.72
21 ConfigNet.SecondDNS=9.181.2.101
22
@echo on
```

Лістинг 5.4. Індивідуальні параметри користувача у файлі завдання розгортання VM.

```
@echo off
1
2 # Каталог с конфигурационными данными
пользователя. Содержимое этого каталога
3 # упаковывается в файл ISO с меткой VMCONFIG и
монтируется как CD-ROM VM.
4 UserConfigDataDir=/root/mydata/
5 # Приложение, которое необходимо запустить
после загрузки системы и настройки сетевых
6 # параметров. Необходимые приложения также
упаковываются в файл VMCONFIG ISO.
UserConfigApp=/root/myapp.sh
@echo on
```

Лістинг 5.5. Варіанти реєстрації і запуску VM у файлі завдання розгортання.

```
@echo off
1
2 # Регистрировать или нет VM на локальном
сервере vmware. Yes - регистрировать, false -
```

```
# не регистрировать. По умолчанию - No.
3 RegisterVM=Yes
5 # Параметры регистрации VM. Для VMware Server
1.x указывать не нужно.
6 RegisterVM.HostURL=https://localhost:8333
7 RegisterVM.Username=root
8 RegisterVM.Password=password
# Включать или нет VM после создания. Yes -
включать, No - не включать.
9 # Default value is false.
PowerOn=Yes
10
@echo on
```

### **5.3 Зміст звіту**

Після закінчення практичного заняття кожен здобувач вищої освіти повинен скласти звіт, який має містити титульний аркуш, мету заняття, хід роботи зі скринами (Print Screen) зроблених завдань, висновки про виконану роботу. Звіт оформляє кожен здобувач вищої освіти індивідуально.

### **Контрольні запитання**

- 1 Платформа як сервіс (PaaS).
- 2 Програмне забезпечення як сервіс (SaaS).
- 3 Безпека.
- 4 Перешкоди розвитку хмарних технологій.
- 5 Розподілені обчислення (grid computing).
- 6 Порівняльний аналіз моделей хмарних технологій.

## **Практичне заняття 6. Модель обслуговування SAAS. Розгортання хмарного сховища OWNSKLOUD**

**Мета роботи:** ознайомитися з основними моделями надання хмарних послуг (SaaS, PaaS, IaaS), розгорнути власне хмарне сховище, доступне користувачам у межах локальної мережі.

### **6.1 Теоретичні основи хмарних сервісів**

Хмарні обчислення («хмари», Scloud computing) – модель надання обчислювальних ресурсів, що охоплює все, від додатків і до центрів обробки даних (ЦОД), через Інтернет за умови оплати за фактичне використання. Однією з перших, хто став упроваджувати послугу хмарних обчислень, стала компанія Amazon, у той час (2002 рік) вона ще була книжковим інтернет-магазином, який згодом переріс завдяки цим упровадженням в одну з найпотужніших технологічних компаній [1–7].

Уже у 2006 році був запущений проєкт під назвою Computing Scloud (Amazon EC2) 1, після цього у 2009 році компанія Google представила Google Apps2. Після цих подій були сформовані загальні поняття про хмарні обчислення, зокрема виділені найбільш важливі моделі обслуговування і моделі розгортання. Хмарні обчислення є наступним кроком в еволюції архітектури побудови інформаційних систем. Завдяки перевагам цього підходу цілком очевидно, що багато інформаційних систем найближчим часом переносяться або будуть перенесені у хмару. Процес уже йде повним ходом і його ігнорування або недооцінювання може призвести до поразки в конкурентній боротьбі на ринку.

Розрізняють три основні моделі обслуговування:

1) програмне забезпечення (ПЗ) як послуга (Software as a Service, SaaS);

- 2) платформа як послуга (Platform as a Service, PaaS);
- 3) інфраструктура як послуга (Infrastructure as a Service, IaaS).

Такі моделі обслуговування, як Container as a Service (CaaS) і Database as a Service (DaaS), є переважно окремими випадками IaaS і SaaS відповідно.

SaaS – модель, при якій не потрібно купувати, встановлювати, оновлювати і підтримувати ПЗ, це завдання бере на себе постачальник послуги. Крім того, здійснити реєстрацію і використання хмарних додатків можна негайно, додатки і дані додатків доступні з будь-якого пристрою, підключеного до інтернету. При поломці пристрою дані не втрачаються, вони зберігаються у хмарі, користувачеві, як правило, доступні локальні налаштування конфігурації програми.

Прикладами моделі SaaS можуть служити поштова служба Gmail, Dropbox, Google Docs, Microsoft Office 365.

Моделі розгортання хмари можна поділити на чотири види (рисунок 6.1).

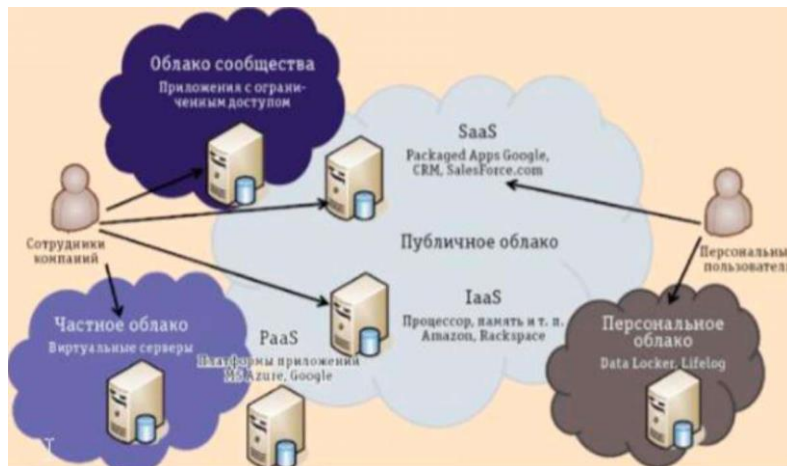


Рисунок 6.1 – Відмінності між моделями розгортання хмари

Приватна хмара призначена для використання однією організацією, як правило, вона перебуває у власності самої організації.

Публічна хмара призначена для широкої публіки, як правило, вона перебуває у власності сторонніх організацій.

Громадська хмара, як правило, призначена для спільноти або організації, а гібридна хмара складається з двох або більше різних хмарних інфраструктур [1–7].

Основними перевагами використання хмарних технологій є:

- зниження витрат на закупівлю обладнання і побудову ЦОД;
- зручність використання додатків з великої кількості пристроїв, зокрема мобільних;
- забезпечення надійності зберігання даних, продуктивності додатків, за рахунок простоти використання ПЗ, моніторингу, балансування навантаження, міграції даних тощо;
- вільний вебдодаток, призначений для синхронізації даних між сервером і клієнтами, як правило, даними є документи і медіаконтент.

OwnScloud є альтернативою таким хмарним сервісів, як Dropbox, Google Drive, Яндекс Диск, MEGA та ін.

Відмінність полягає в тому, що додаток можна розгорнути на власному сервері як для домашнього використання, так і використання в організаціях. Оскільки додаток поширюється безкоштовно, має відкритий вихідний код і періодично обростає новими функціями (планувальник завдань, календар, фотогалерея, переглядач документів, гнучка автентифікація користувачів та ін.), проєкт набув популярності як у користувачів, так і Open-source розробників [1–7].

## **6.2 Порядок виконання роботи**

Для виконання практичного заняття як сервер може використовуватися ВМ, встановлена на практичному занятті 1.



- 1 Установити дистрибутив Debian. ВМ повинна мати доступ до мережі Інтернет для скачування потрібних пакетів (додаток В).
- 2 Установити ownCkcloud Server у ВМ (додаток Г).
- 3 На хост-машині або іншому комп'ютері мережі завантажити і встановити ownCkcloud Client.
- 4 Підключити ownCkcloud Client до сервера під час першого запуску клієнта, а також синхронізувати всі дані з сервера (додаток Д).
- 5 Перевірити роботу синхронізації даних між клієнтом і сервером (створити, видалити, перемістити файл або каталог).
- 6 Надати публічне посилання на файл або каталог і перевірити його доступність у межах локальної мережі.
- 7 Ознайомитися з додатковими можливостями ownCkcloud.
- 8 Підготувати звіт з практичного заняття, у якому описати результат виконаної роботи, вивчені особливості ownCkcloud платформи, а також порівняння ownCkcloud з іншими SaaS сервісами.
- 9 Проаналізувати, які переваги і недоліки має ownCkcloud порівняно з іншими хмарними постачальниками.
- 10 Порівняти ownCkcloud з іншими хмарними сховищами, такими як Google Drive, DropBox, Amazon S3, Amazon Ckcloud Drive, OneDrive [8–14].

### **6.3 Зміст звіту**

Після закінчення практичного заняття кожен здобувач вищої освіти повинен скласти звіт, який має містити титульний аркуш, мету заняття, хід роботи, скрини (Prnt Scrn) зроблених завдань і висновки про виконану роботу. Звіт оформлює кожен здобувач вищої освіти індивідуально.

## **Контрольні запитання**

- 1 Організація безпеки хмарних технологій.
- 2 Зниження витрат на програмне забезпечення.
- 3 Підвищення гнучкості і швидкості реагування системи.
- 4 Чи можуть несумісні додатки працювати на одному комп'ютері?
- 5 Віртуалізація застосувань.
- 6 Сучасний стан і проблеми в GRID.
- 7 Сучасні тенденції розвитку GRID-систем.
- 8 Web 2.0/3.0 та GRID.

## **Практичне заняття 7. Модель обслуговування PaaS на прикладі деплоя додатка в HEROKU**

**Мета роботи:** ознайомитися з моделлю гнучкої розробки програмного забезпечення, розгорнути додаток на Heroku, ознайомитися з системою контролю версій Git.

### **7.1 Теоретичні основи хмарних сервісів**

Heroku є хмарною PaaS-платформою, що підтримує ряд мов програмування: Java, Node.js, Scala, Clojure, Python і PHP. У Heroku можна легко розгорнути (деплоїти) проєкти, не турбуючись про розгортання власної інфраструктури для однієї програми [1–7].

Програми, що працюють на Heroku, використовують власний DNS-сервер Heroku. Для кожної програми виділяється кілька незалежних віртуальних процесів, які називаються «dynos». Процеси розподілені за спеціальною віртуальною сіткою, що складається з декількох серверів.

Heroku підтримує систему контролю версій Git, а також підключення до акаунту GitHub [1–7].

### **7.2 Порядок виконання роботи**

- 1 Зареєструватися в Heroku (додаток Е).
- 2 Створити тестовий додаток на node js (або будь-якому іншому мовою технології).
- 3 Задеплоїти додаток на Heroku.
- 4 Внести зміни до початкового коду програми та знову задеплоїти додаток.
- 5 Ознайомитися з додатковими можливостями Heroku і Git.

6 Підготувати звіт з практичного заняття, у якому описати результат виконаної роботи, вивчені особливості PaaS платформи Heroku, а також порівняння Heroku з іншими PaaS сервісами.

7 Проаналізувати, які переваги і недоліки має Heroku порівняно з іншими хмарними постачальниками.

8 Порівняти Heroku з іншими хмарними постачальниками, такими як Microsoft Azure, Google App, IBM Bluemix, Oracle Cloud Platform, OpenShift Red Hat [8–14].

### **7.3 Зміст звіту**

Після закінчення практичного заняття кожен здобувач вищої освіти повинен скласти звіт, який має містити титульний аркуш, мету заняття, хід роботи, скрини (Print Screen) зроблених завдань і висновки про виконану роботу. Звіт оформлює кожен здобувач вищої освіти індивідуально.

### **Контрольні питання**

- 1 Перешкоди розвитку хмарних технологій.
- 2 Моделі та технології організації.
- 3 Workflow у гетерогенному середовищі хмара-Grid.
- 4 Azure Service Bus.
- 5 Платформа як Сервіс (PaaS).
- 6 Microsoft Azure.
- 7 Програмне забезпечення як Сервіс (SaaS).
- 8 Комунікація як Сервіс (CaaS).

## **Практичне заняття 8. Використання моделі обслуговування PAAS для виконання розрахунково-графічної роботи**

**Мета роботи:** закріпити навички, отримані на попередніх заняттях, і самостійно виконати індивідуальне розрахунково-графічне завдання.

### **8.1 Теоретичні основи**

Розрахунково-графічна робота (РГР) – індивідуальне завдання, яке передбачає розв'язання конкретної практичної навчальної задачі з використанням відомого, а також (або) самостійно вивченого теоретичного матеріалу. Певну частину такої роботи складає графічний матеріал, який виконується відповідно до чинних нормативних вимог [1-7].

Розрахунково-графічна робота – це завдання на індивідуальну роботу здобувачу вищої освіти за індивідуальним варіантом. Роботу виконують на аркушах А4. Виконання цієї роботи має дві мети. Перше і найголовніше – можливість потренуватися у розв'язанні задач. Тому робити всі вправи необхідно самостійно. Якщо виникнуть труднощі, здобувач вищої освіти може звернутися до викладача. Друга мета – контроль роботи з боку викладача. Розрахунково-графічну роботу після перевірки потрібно буде захищати, тобто продемонструвати, що здобувач вищої освіти дійсно справився з поставленим завданням.

Ще одне зауваження щодо термінів виконання розрахунково-графічних робіт. Викладач, даючи завдання, оголосить кінцевий термін здачі роботи на перевірку. Але робити приклади треба одразу, як тільки прочитають відповідну тему. Тоді здобувач досягне основної мети – роблячи приклади, набувати відповідних навичок.

## **8.2 Порядок виконання роботи**

Виконати роботу самостійно [8–14].

Розрахунково-графічна робота – це вид самостійної роботи здобувачів вищої освіти, що виконується з метою систематизації та закріплення знань і умінь, отриманих при вивченні навчального матеріалу окремих тем, розділів дисципліни. Структурними елементами розрахунково-графічної роботи є тема, теоретична частина, де наведено стислий виклад теоретичного матеріалу за певною темою; практична частина,

Порядок виконання розрахунково-графічної роботи і результати фіксуються письмово.

## **8.3 Зміст звіту**

Після закінчення практичного заняття кожен здобувач вищої освіти повинен скласти звіт, який має містити титульний аркуш, мету заняття, хід роботи, скрини (Print Scrn) зроблених завдань і висновки про виконану роботу. Звіт оформлює кожен здобувач вищої освіти індивідуально.

### **Контрольні запитання**

- 1 Моніторинг як Сервіс (MaaS).
- 2 GoogleApps.
- 3 Функції, доступні користувачеві.
- 4 App EngineApp Engine.

## 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. Understanding 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.



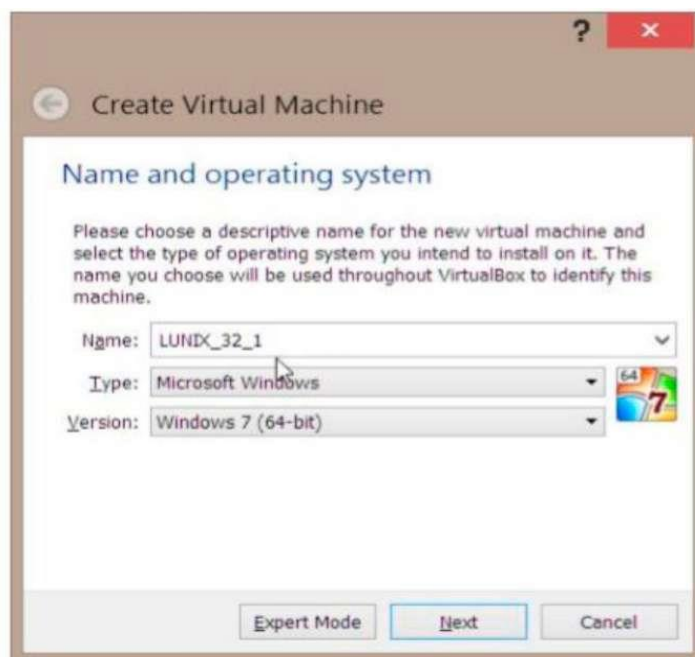
## ДОДАТОК А

### Створення віртуальної машини у VirtualBox

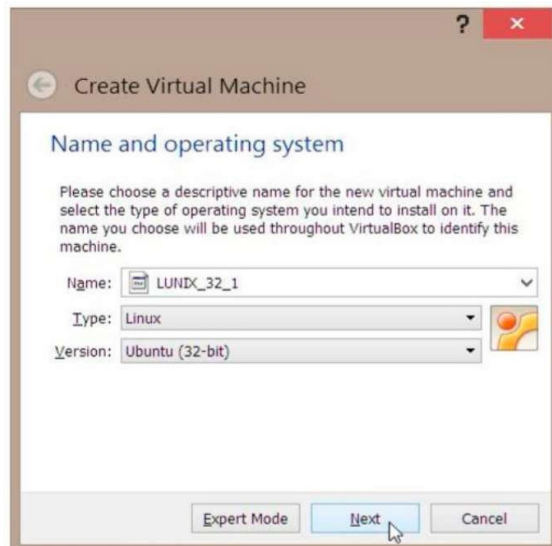
1 Запустіть інсталятор VirtualBox. У вікні є можливість управляти ВМ. Для цього необхідно створити ВМ (кнопка New).



2 У полі Name дайте ім'я ВМ.

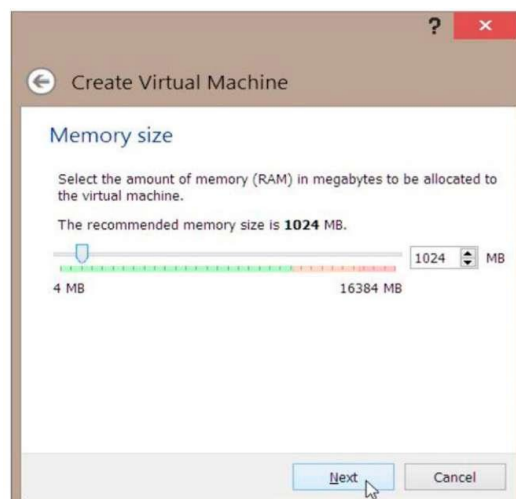


3 Оскільки ми будемо встановлювати Linux як ОС, слід явно це вказати при створенні. Зробити це можна за допомогою полів Type і Version. Натисніть Next.



4 На наступному етапі необхідно визначити, скільки оперативної пам'яті робочої станції допустимо віддати в розпорядження віртуальній машині – чим більше виділимо, тим менше ресурсів залишиться у host комп'ютера для основних завдань, але тим легше буде працювати віртуальній машині.

Для нескладних завдань віртуальній машині вистачить 1024 Мбайт, як і пропонує нам сам VirtualBox. Цікавий факт: оперативна пам'ять повернеться нашому комп'ютеру, коли VM буде виключена. Тобто у вас може бути, наприклад, 10 машин, кожна з яких буде забирати 1 Гбайт, але якщо всі вони будуть вимкнені, ваш комп'ютер буде мати всю оперативну пам'ять знову у своєму розпорядженні.



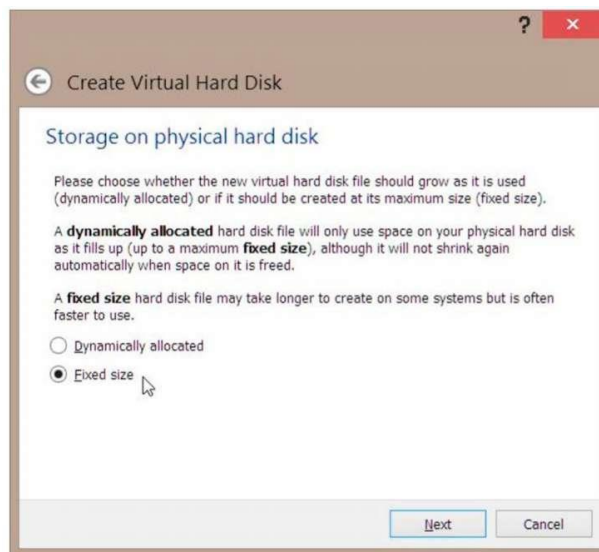
5 У наступному вікні необхідно вибрати, скільки пам'яті на жорсткому диску буде доступно гостьовій OS, щоб зберігати свої файли. Натискаємо Next.



6 За замовчуванням вибираємо VDI тип диска.



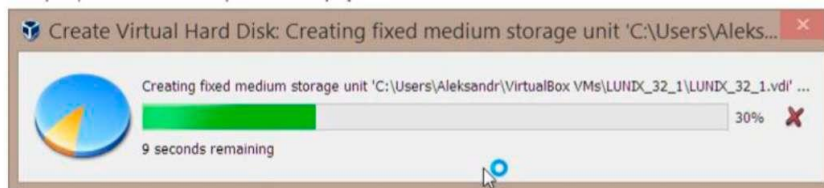
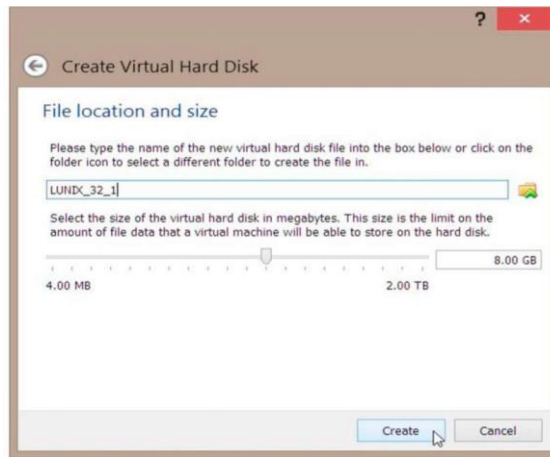
7 Тут варіант Fixed Size може працювати швидше, тому вибираємо його.



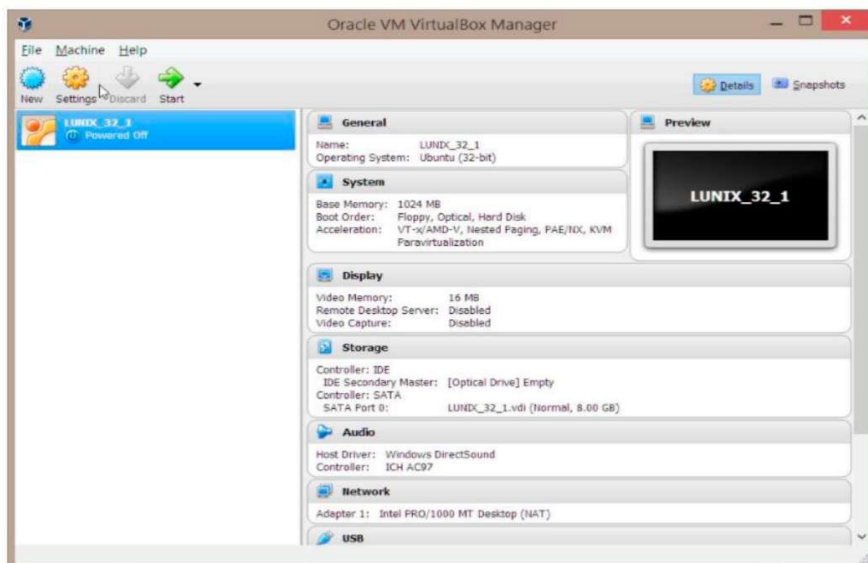
8 На наступному екрані VirtualBox пропонуємо віддати VM 8 Гбайт жорсткого диска. Зменшимо до 4 Гбайт. Перед установленням зайдіть у Мій Комп'ютер і переконайтеся, що у вас є стільки вільного місця, перш ніж натиснути Next.

Ще один цікавий факт: пам'ять на жорсткому диску – це великий файл, у якому VM зберігає свої файли. І навіть коли вона вимкнена, місце на жорсткому диску вашого комп'ютера все одно зайнято збереженими файлами, щоб ви могли з ними працювати, коли забажаєте цього. Тому варто пам'ятати, що доступний вам обсяг жорсткого диска скоротиться, а якщо VM у вас кілька, кожна виділить місце для себе окремо.

9 Процес створення файлів VM.



10 VM створена. Вона доступна в менеджері операційних систем, її властивості докладно розписані справа.



## ДОДАТОК Б

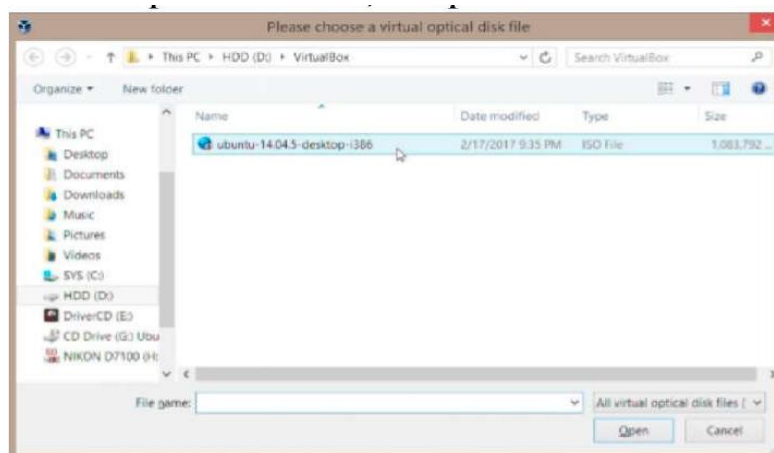
### Встановлення операційної системи Ubuntu у VirtualBox

1 Обираємо у VirtualBox меню налаштувань (Settings).

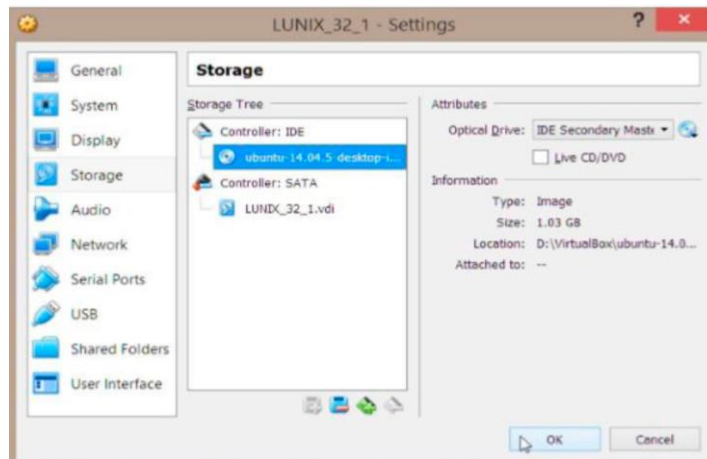
2 У вікні, що з'явилося, вибираємо зліва пункт Storage, рядок Empty в розділі IDE. Це аналог порожнього привода DVD, у який можна вставити диск-образ – образ дистрибутива Linux. Після натискання на Empty необхідно натиснути на іконку диска.



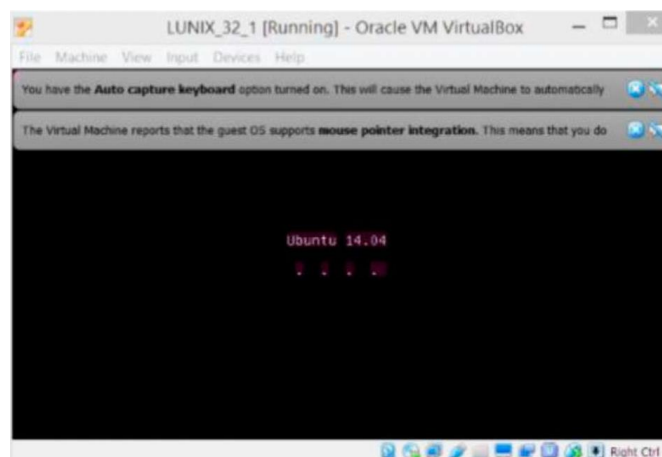
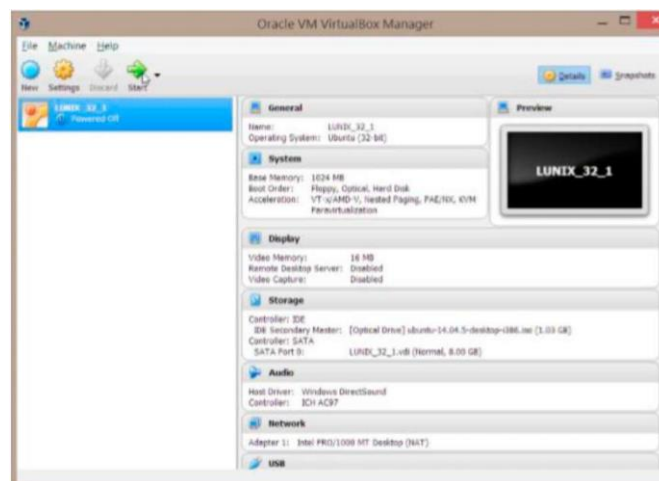
3 Обираємо файл із образом диска, відкриваємо.



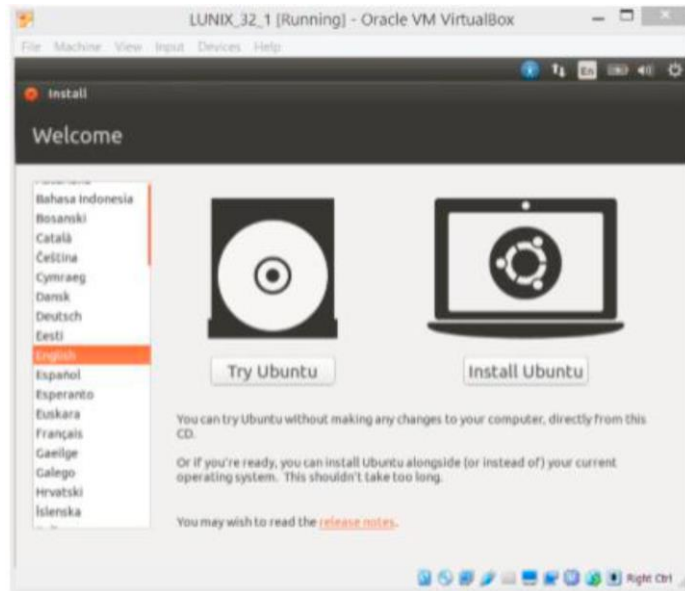
4 Тепер віртуальний DVD-привод уже не порожній (Empty), а підключений до образу Linux.



5 VM готова до увімкнення. При запуску вона визначить, що ОС вона не має, але в її віртуальному приводі є диск Linux. І завантажить його. Натисніть Start.



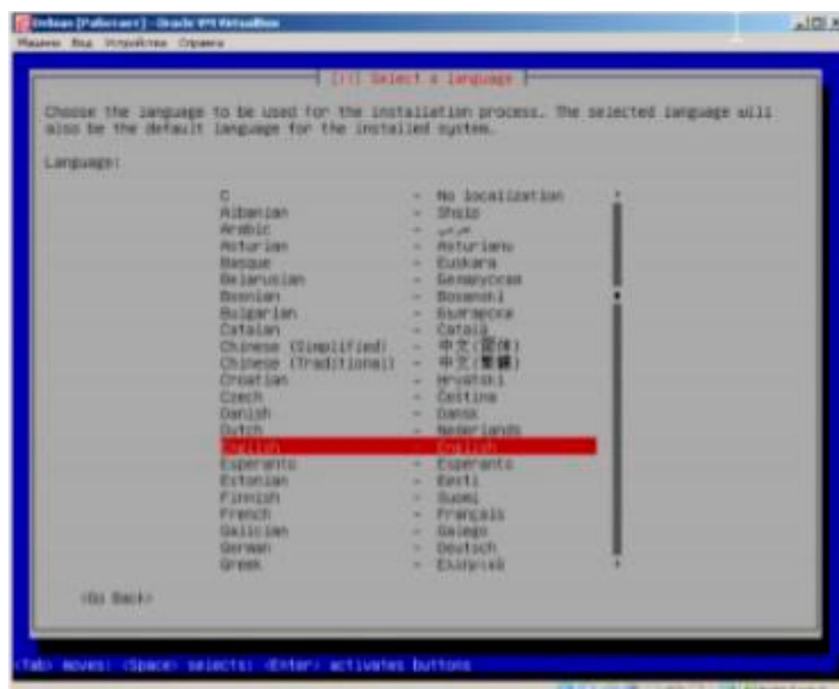
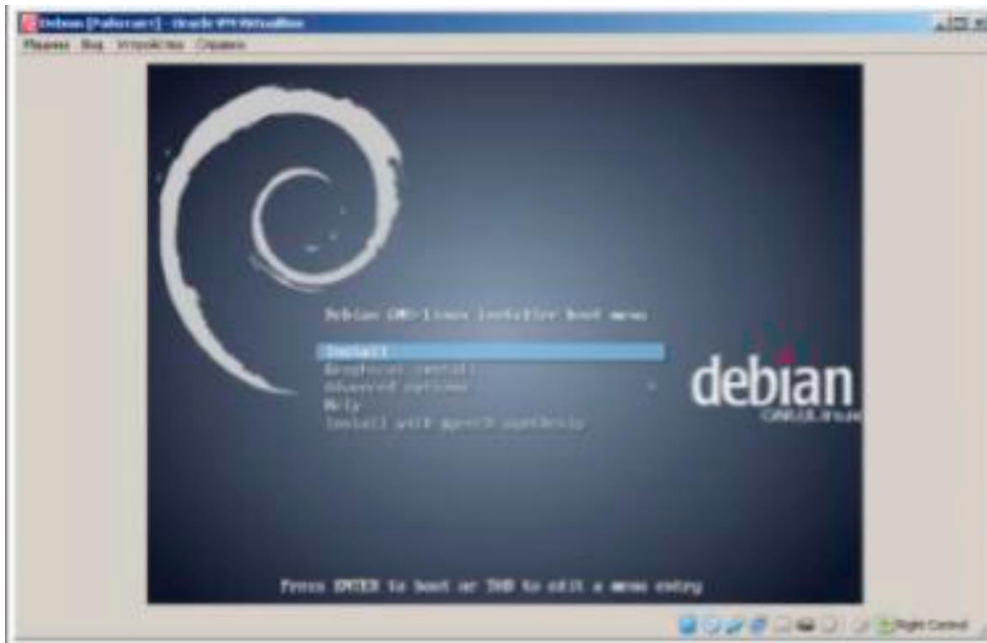
6 Після завершення завантаження виберіть Try Ubuntu.

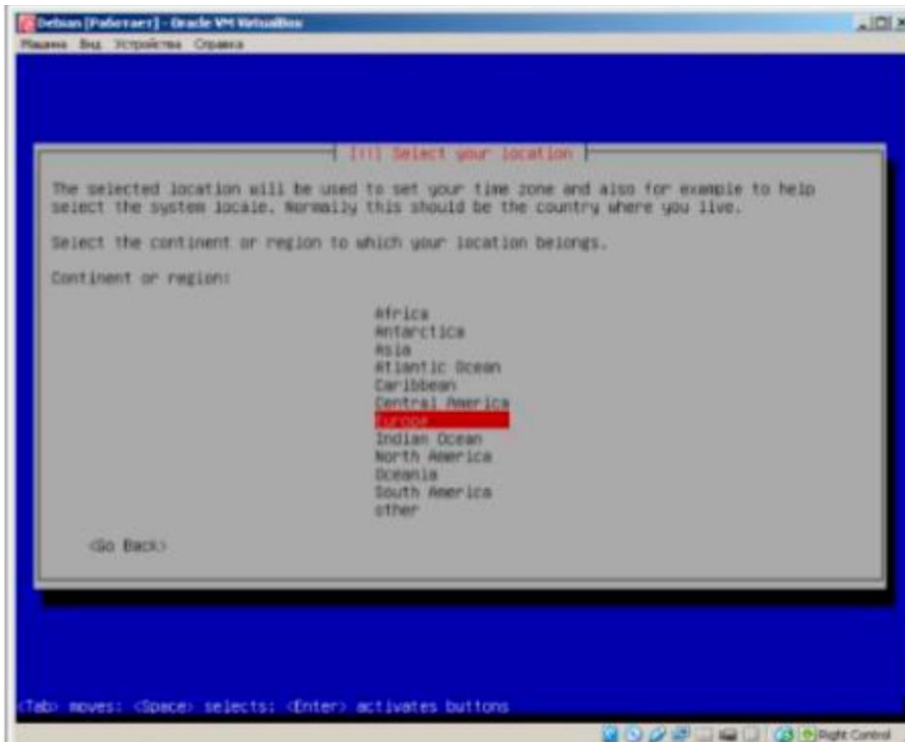
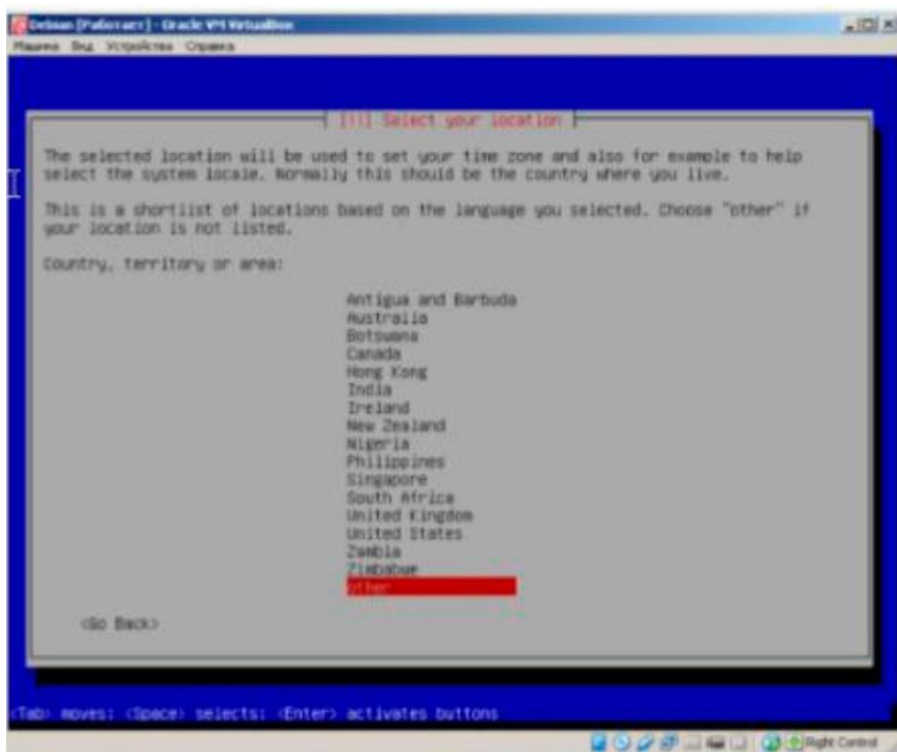


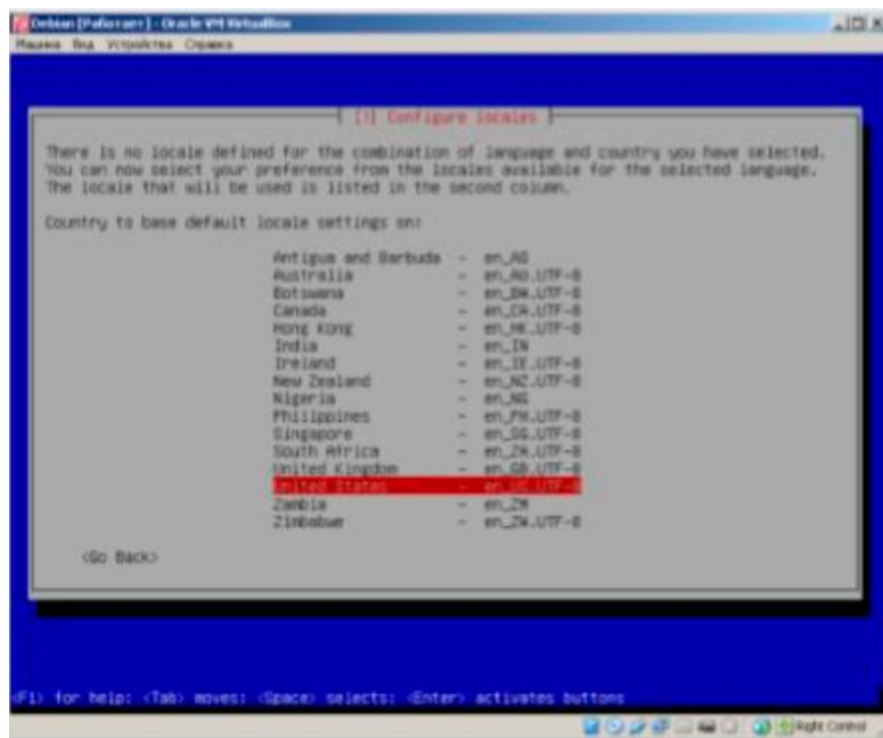
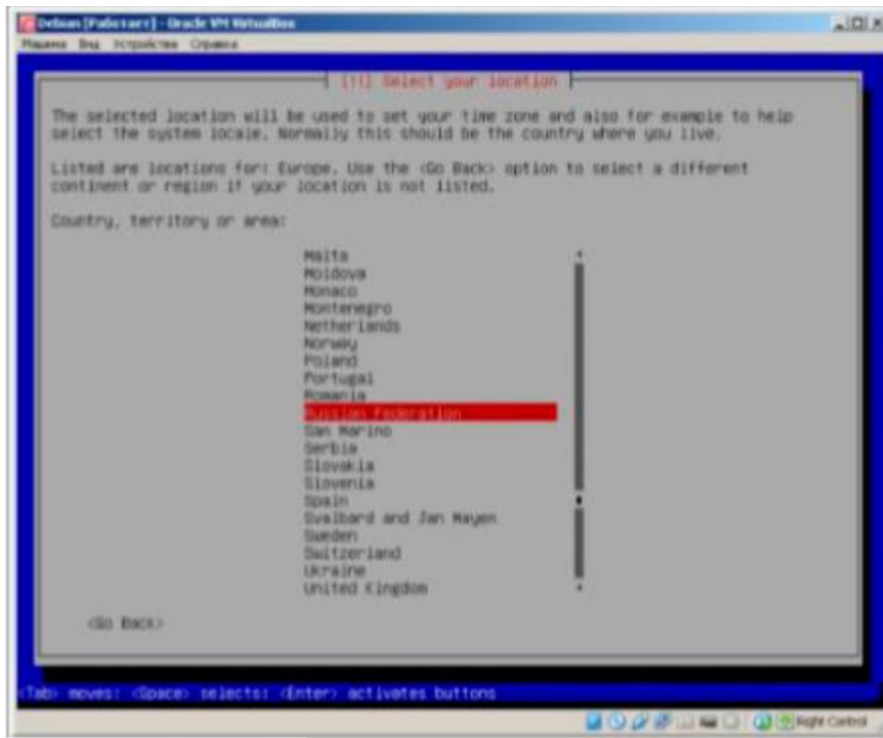


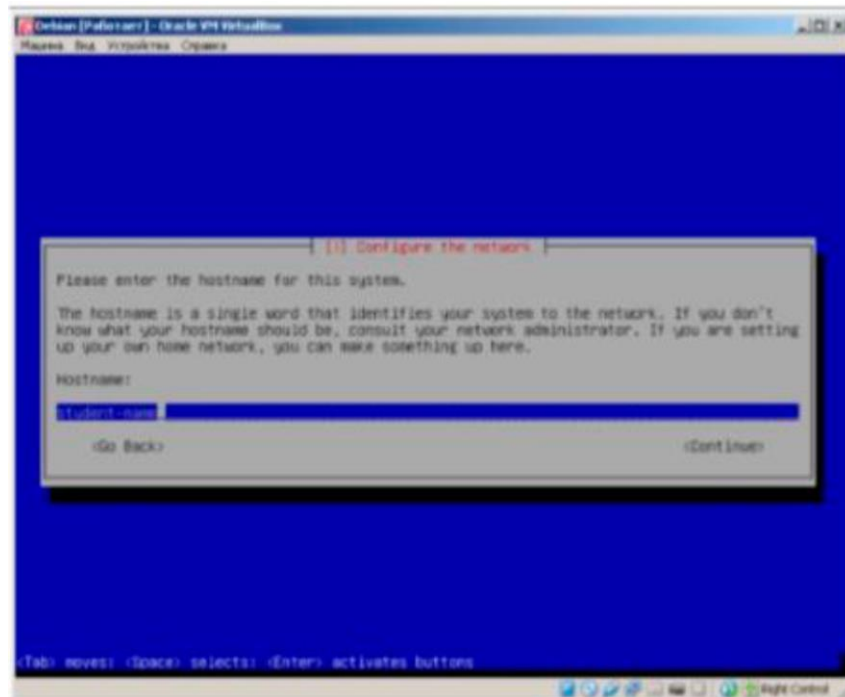
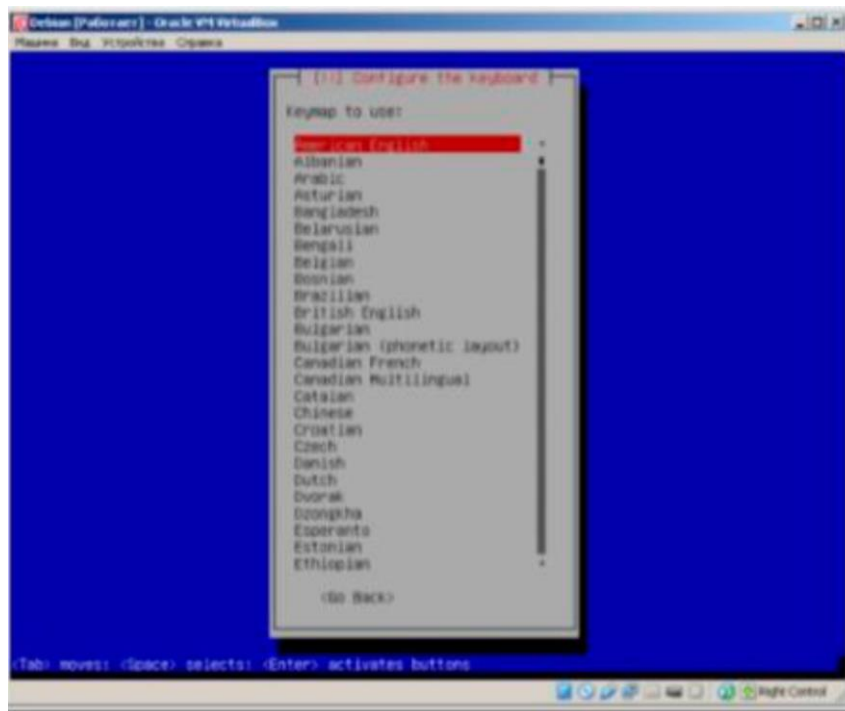
# ДОДАТОК В

## Встановлення Debian GNU / Linux у VirtualBox

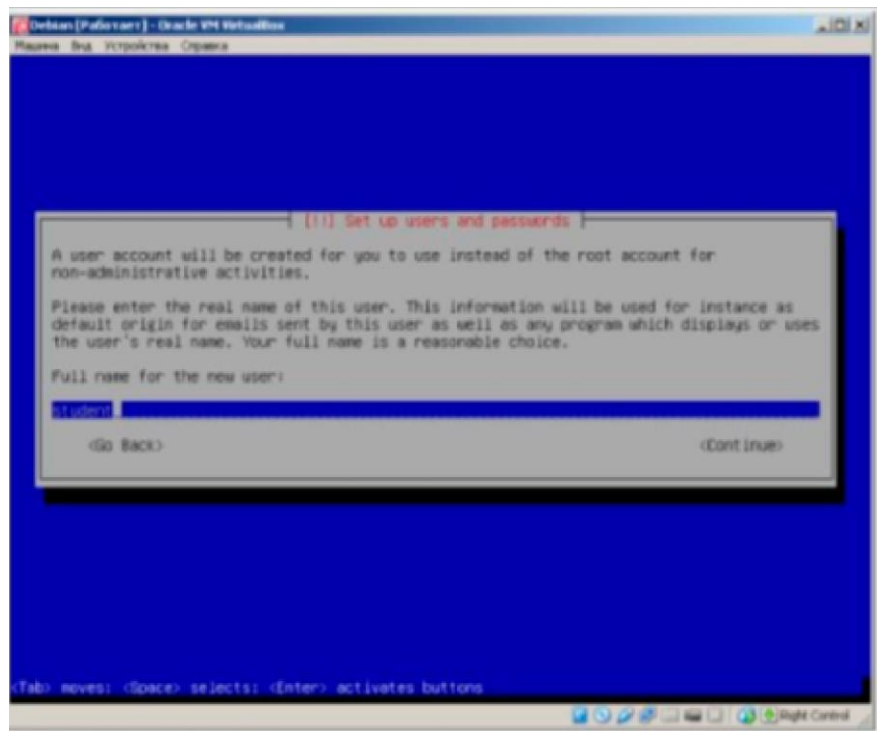
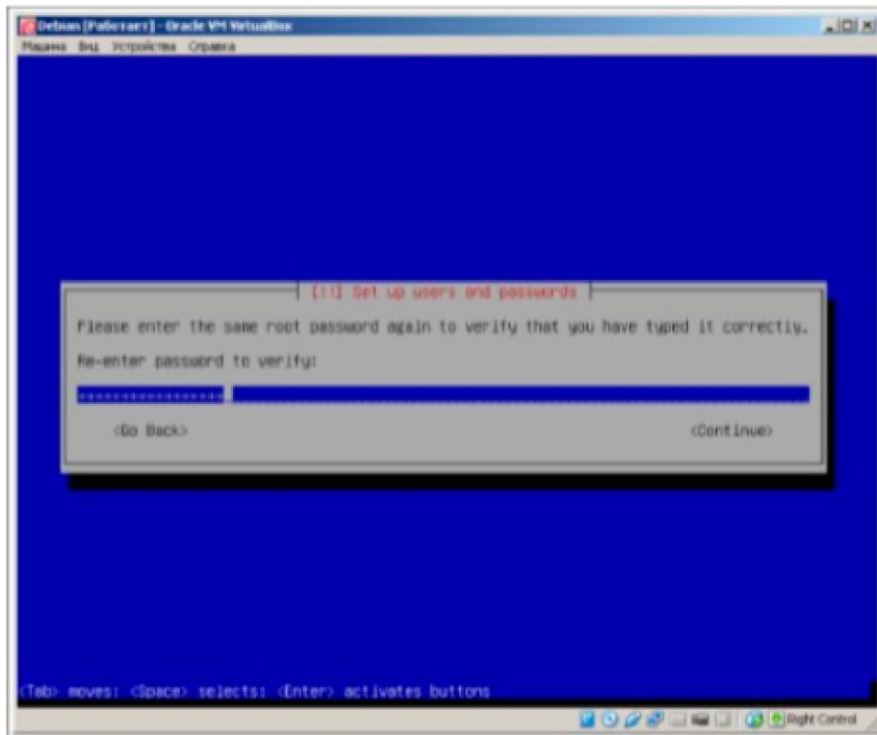


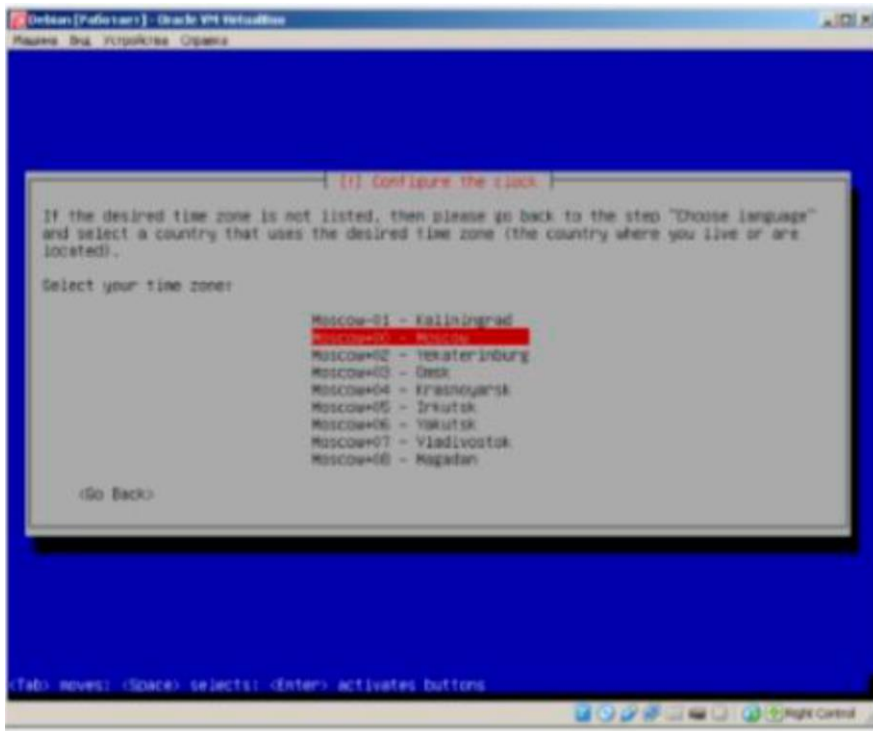
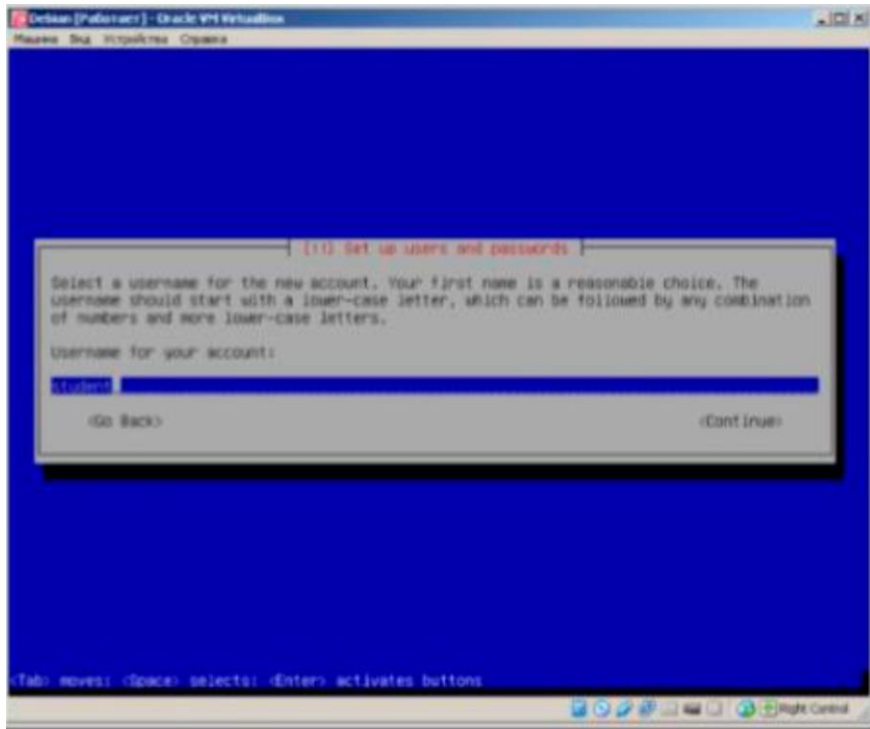


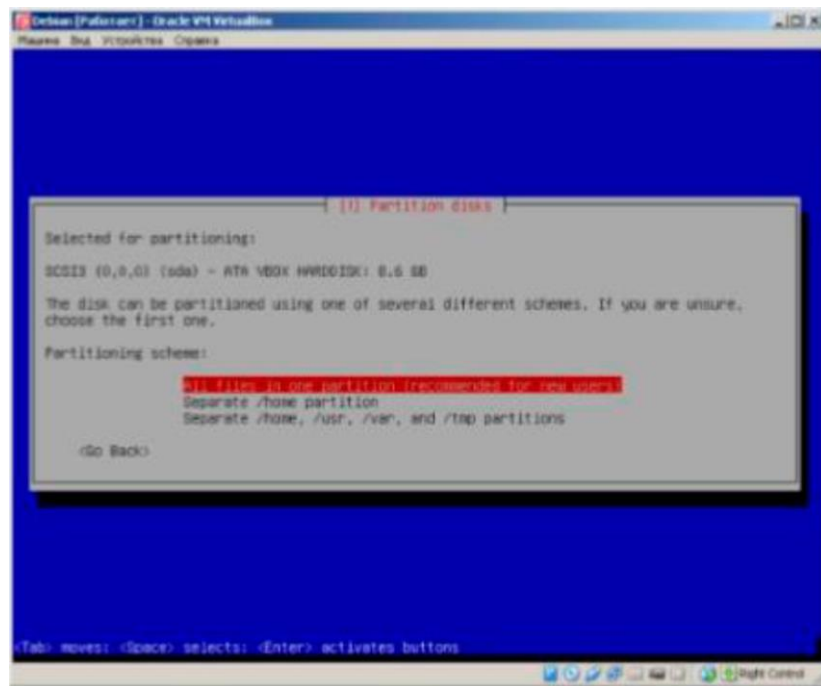
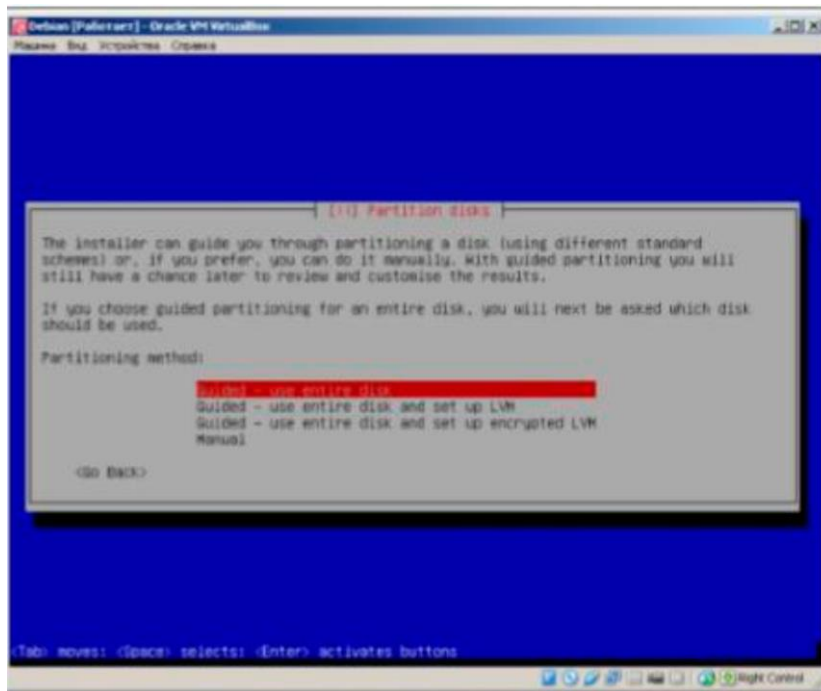




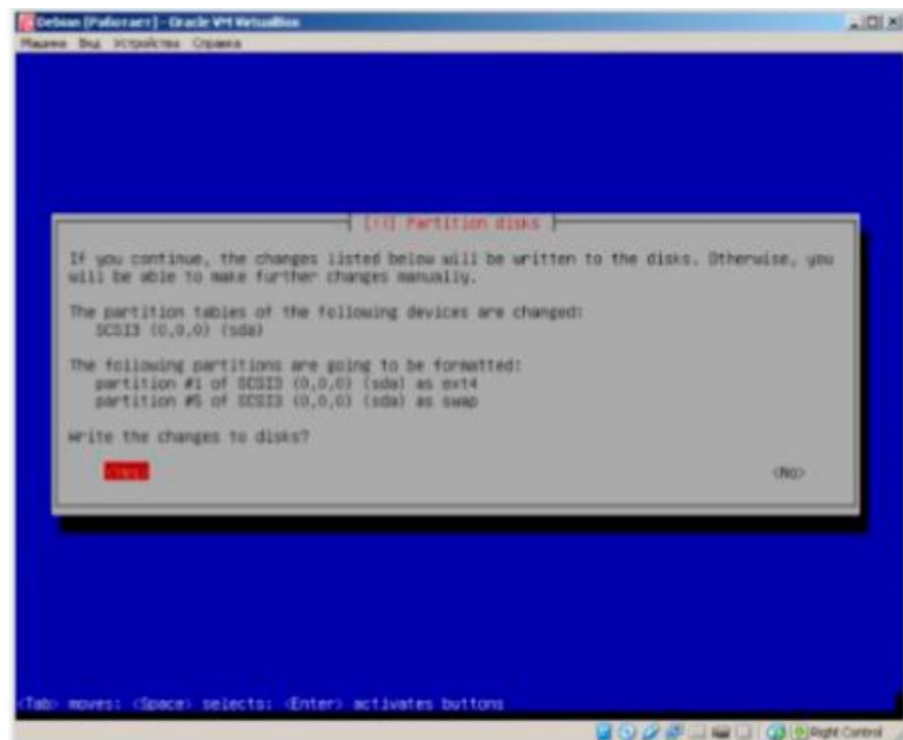
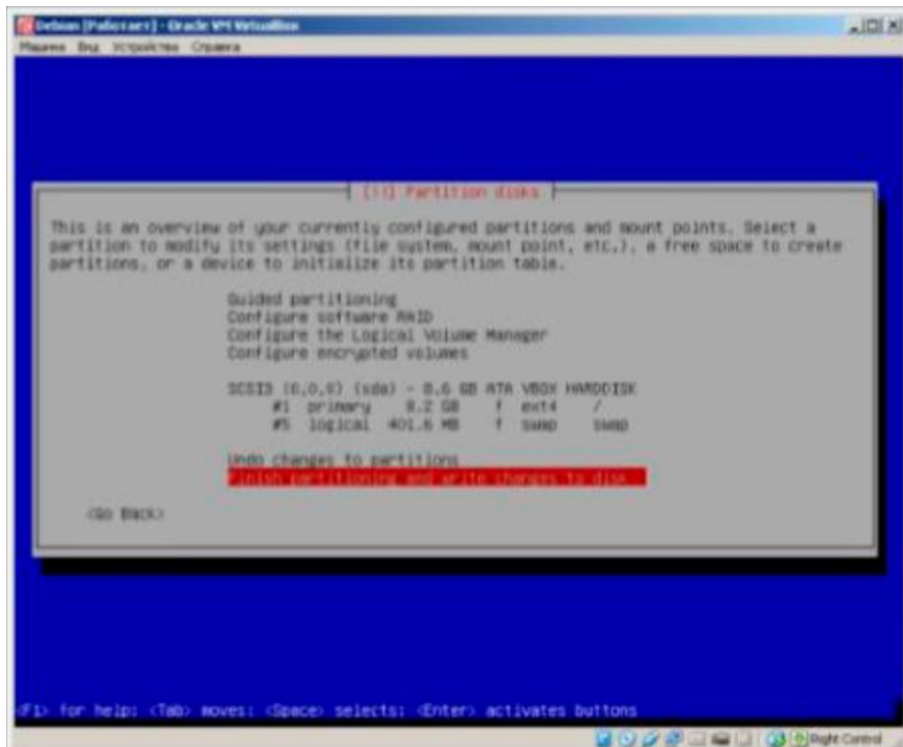


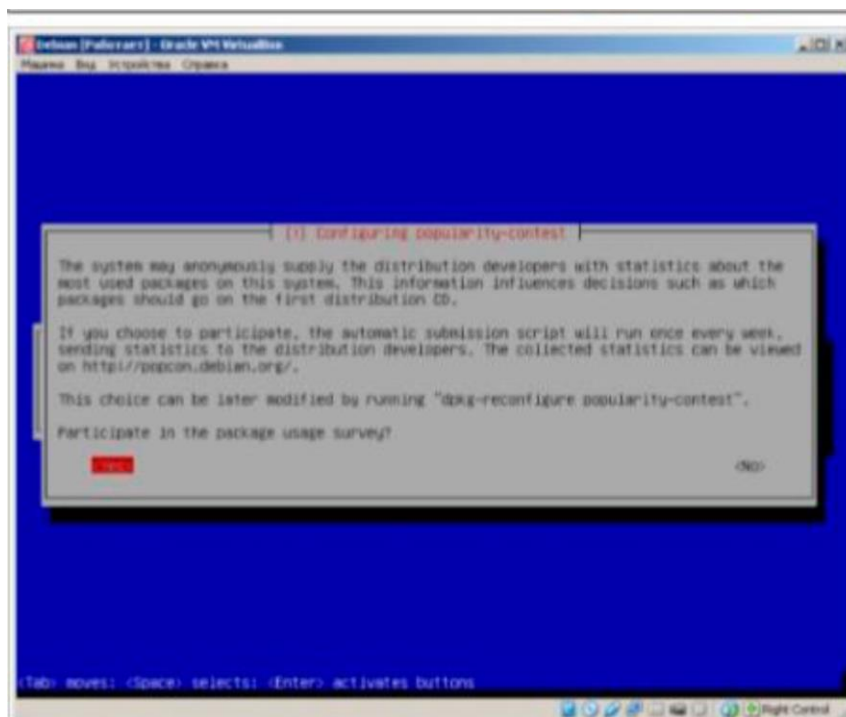
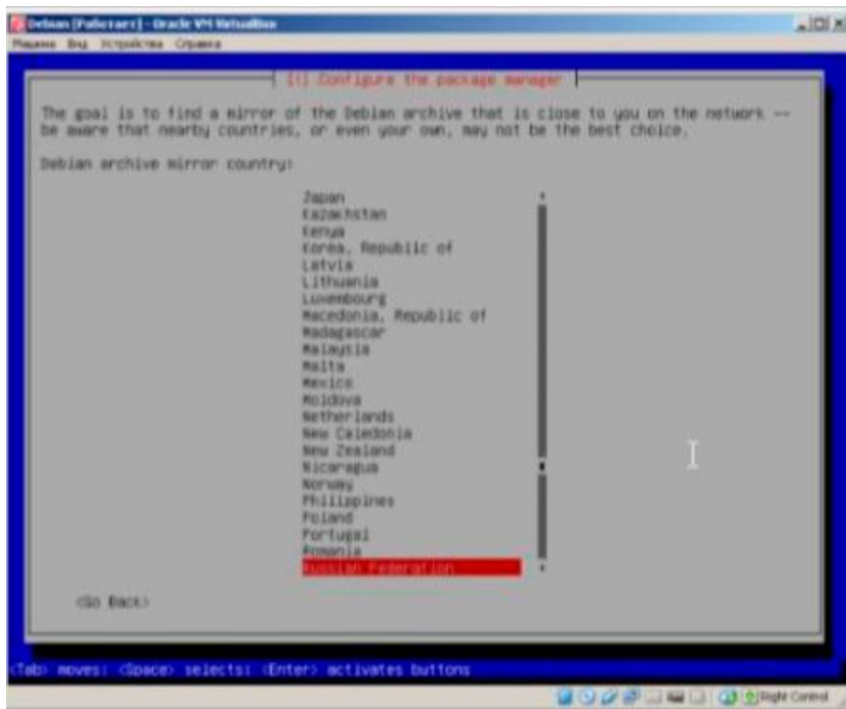








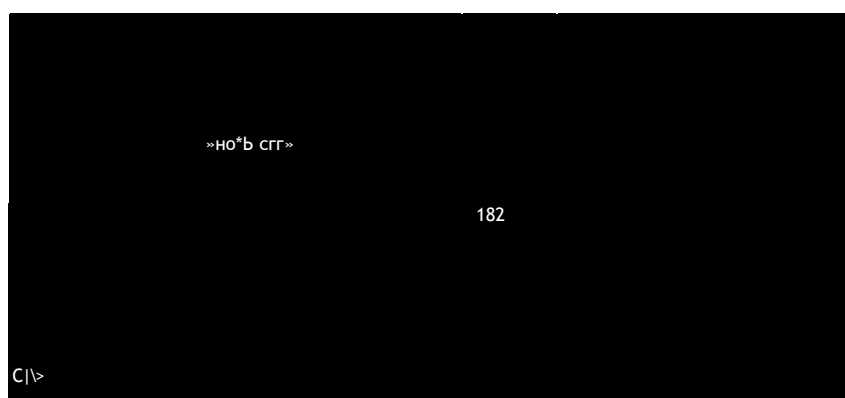






Якщо після встановлення операційна система не отримує автоматичних налаштувань мережі через DHCP, то необхідно налаштувати мережу вручну. Насамперед необхідно переконатися, що в мережевого адаптера в налаштуваннях віртуальної машини виставлено режим «Мережевий міст». Потім за допомогою команди `ipconfig` у Windows або `ifconfig` у Linux/MacOS необхідно дізнатись параметри своєї мережі на хост-комп'ютері.

Налаштування мережі на хост-комп'ютері з Windows.



```
»но*ь сгг»  
182  
C:\>
```

Тут видно, що хост-комп'ютер перебуває в мережі 192.168.0.0, має адресу 192.168.0.10, шлюзом є маршрутизатор із IP-адресою 192.168.0.1. Налаштуємо мережевий інтерфейс для віртуальної машини вручну, нехай адреса віртуальної машини буде 192.168.0.102. Для цього необхідно відредагувати конфігураційний файл мережі та привести його до такого вигляду: `# nano /etc/network/interfaces`.

Додати до файлу:

```
source /etc/network/interfaces.d/* auto lo  
iface lo inet loopback auto eth0  
iface eth0 inet static  
address 192.168.0.102  
netmask 255.255.255.0
```

```
network 192.168.0.0
broadcast 192.168.0.255
gateway 192.168.0.1
dns-nameservers 192.168.0.1 8.8.8.8
```

У цьому конфігураційному файлі наведено налаштування для основного інтерфейсу мережі (eth0). Після збереження змін у конфігураційному файлі необхідно перезапустити демон мережі: `/etc/init.d/networking restart`.

Перевіримо на VM нові налаштування мережі:

```
ip -4 addr show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu
1500 qdisc pfifo_fast
state UP group default qlen 1000
inet 192.168.0.102/24 brd 192.168.0.255 scope
global eth0 valid_lft forever preferred_lft forever
```

Для перевірки роботи мережі пробуємо пропінгувати адресу 192.168.0.102 або з'єднатися до VM ssh.

## ДОДАТОК Г

### Встановлення ownCkcloud Server у Debian GNU/Linux

Після успішного входу в систему насамперед необхідно отримати права суперкористувача та додати користувача student до групи sudo:

```
$ su -  
Password: toor  
apt install sudo  
usermod -a -G sudo student
```

Додавання репозиторію та ключа для ownCkcloud Server:

```
wget -nv https://goo.gl/mmV2ga -O Release.key  
apt-key add - < Release.key OK  
echo deb http://download.ownckloud.org  
/download/repositories/stable/Debian_8.0/ /  
\ > /etc/apt/sources.list.d/ownckloud.list
```

Після додавання репозиторію необхідно оновити список доступного в репозиторіях ПЗ і запустити встановлення ownCkcloud Server (під час встановлення MySQL установник запросить пароль суперкористувача MySQL, він не обов'язково має збігатися з паролем користувача root):

```
apt update  
apt install ownckloud ownckloud-files  
New password for the MySQL "root" user: toor-  
mysql Repeat password for the MySQL "root" user:  
toor-mysql
```

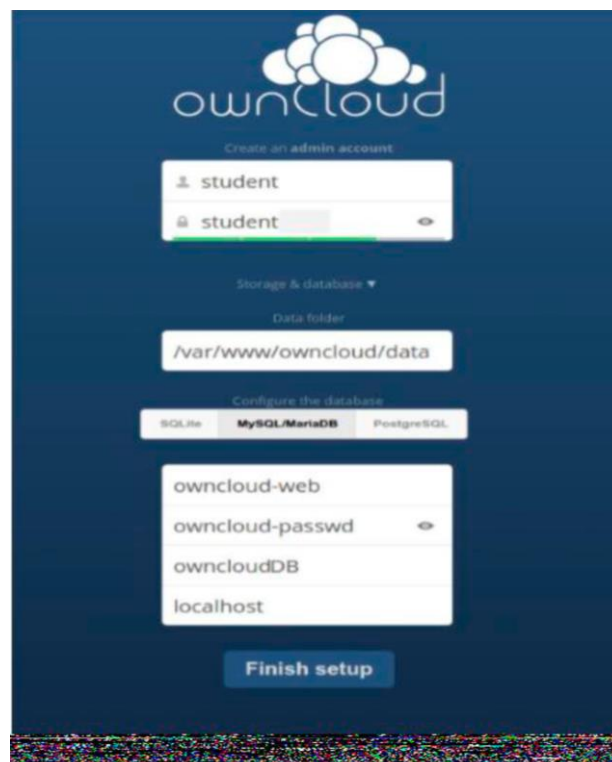
Після того як установник завантажив і встановив усі необхідні пакети, можна перевірити коректність встановлення, зайшовши за адресою [http://192.168.0.102/ownckloud /](http://192.168.0.102/ownckloud/), де 192.168.0.102 — IP-адресу сервера

(ВМ). Програма пропонує використовувати базу даних SQLite2 за замовчуванням, ми для прикладу будемо використовувати MySQL:

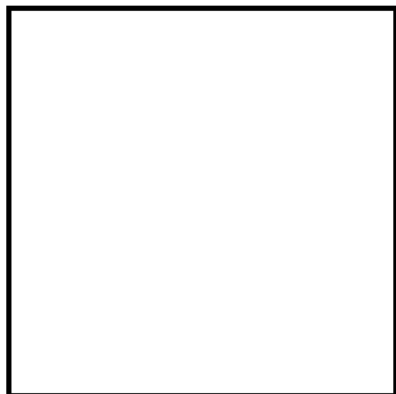
```
mysql -uroot -ptoor-mysql
mysql> CREATE DATABASE ownckloudDB;
mysql> CREATE USER "ownckloud-web"@"localhost" \
-> IDENTIFIED BY "ownckloud-passwd";
mysql> GRANT ALL PRIVILEGES ON ownckloudDB.* \
-> TO "ownckloud-web"@"localhost";
mysql> FLUSH PRIVILEGES;
mysql> quit
```

Після створення бази даних необхідно оновити сторінку програми, налаштувати параметри бази даних і встановити дані облікового запису адміністратора ownCkcloud. Після натискання клавіші Finish Setup база даних і налаштування користувача підключаються до ownCkcloud.

Параметри БД та облікового запису адміністратора.



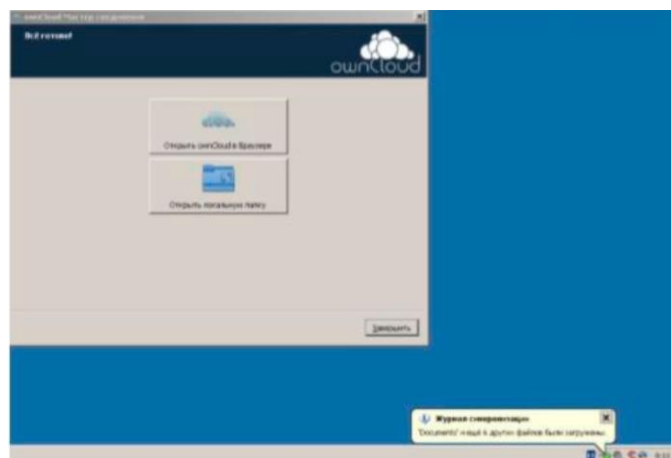
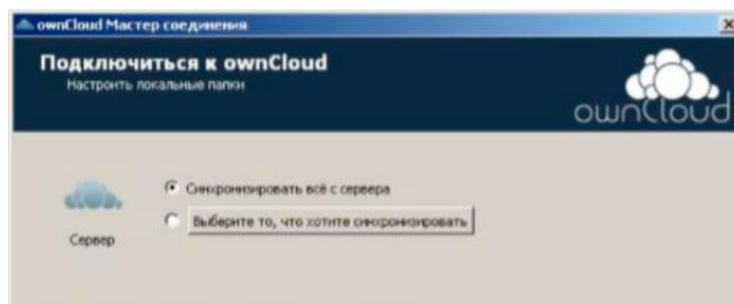
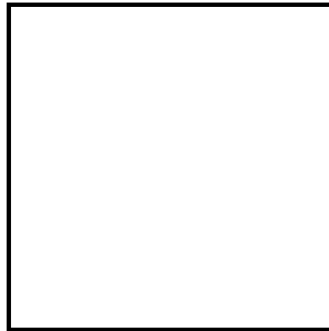
Перший вхід в ownScloud та інтерфейс програми.





# ДОДАТОК Д

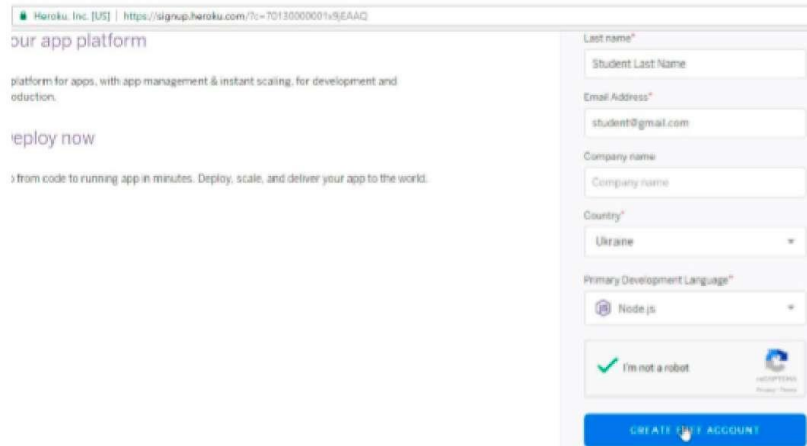
## Підключення ownCloud Client до сервера



## ДОДАТОК Е

### Підключення та реєстрація до Heroku

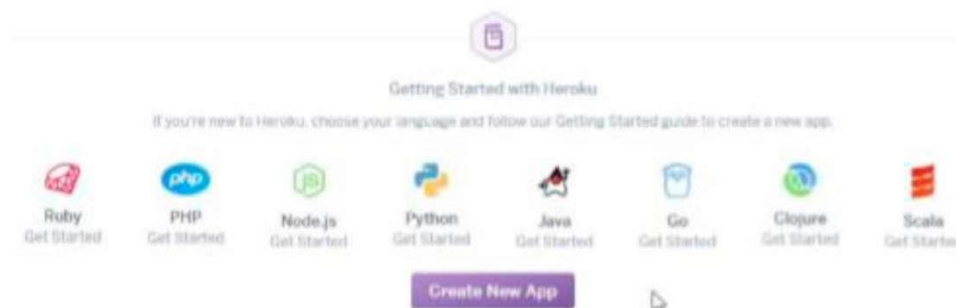
#### Підключення до Heroku



The screenshot shows the Heroku sign-up page. On the left, there is a brief introduction to Heroku as a platform for apps, with app management and instant scaling. On the right, there is a registration form with the following fields: Last name (with a placeholder 'Student Last Name'), Email Address (with a placeholder 'student@gmail.com'), Company name, Country (a dropdown menu with 'Ukraine' selected), and Primary Development Language (a dropdown menu with 'Node.js' selected). Below these fields is a CAPTCHA 'I'm not a robot' and a blue 'CREATE NEW ACCOUNT' button.

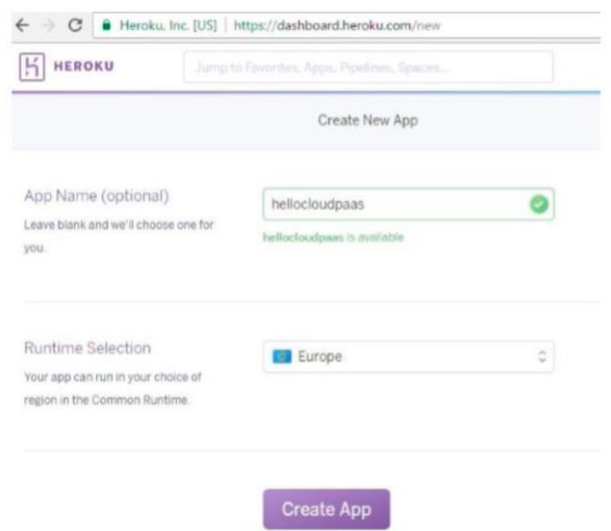
#### Реєстрація в Heroku.

Після реєстрації на вказану поштову скриньку надходить лист із підтвердженням, необхідно перейти за посиланням з листа і підтвердити реєстрацію. Після підтвердження реєстрації можна увійти до свого облікового запису та ознайомитися з інтерфейсом платформи. Створимо нову програму в розділі Personal Apps.



#### Список створених програм у Heroku.

Нехай назва програми буде «hellockloudraas».



Встановити Heroku CLI (див. інсталятори в додатку до практичного заняття). Розпакувати архів `nodejs-docs-samples-master.zip`. Запустити Git Bash. Виконати такі команди: Авторизуємось у Heroku:

```
$ heroku login
```

Додаємо контактні дані розробника:

```
$ git config --global --add user.email  
"student123@gmail.com" $ git config --global --add  
user.name "Name Surname"
```

Перемістимося в каталог з тестовим додатком:

```
$cd appengine/hello-world/
```

Перемістимося в каталог з тестовим додатком:

```
$cd appengine/hello-world/
```

Ініціалізуємо репозиторій:

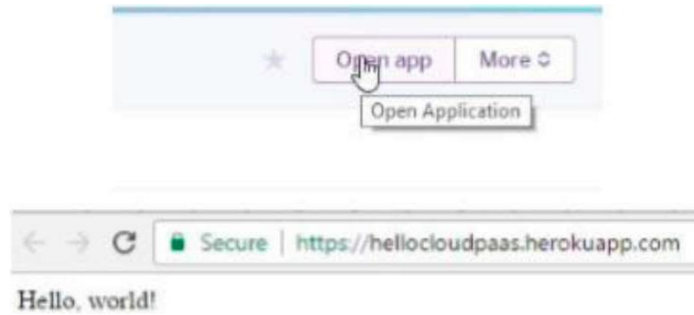
```
$ git init  
$ heroku git:remote -a hellocloudpaas
```

Деплоїмо додаток у Heroku:

```
$ git add .
```

```
$ git commit -am "Some comment" $ git push heroku  
master
```

Відкриваємо програму.



## МЕТОДИЧНІ ВКАЗІВКИ

до практичних занять і розрахунково-графічної роботи з дисциплін

### *«ПАРАЛЕЛЬНІ ТА РОЗПОДІЛЕНІ ОБЧИСЛЕННЯ ТА СКLOUD-ТЕХНОЛОГІЇ»*

Відповідальний за випуск Мірошник М. А.

Редактор Ібрагімова Н. В.

---

Підписано до друку 30.06.2021 р.

Умовн. друк. арк. 5,75. 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.