Адміністрування Unix-подібних операційних систем

Практичний проект №1

НАЗВА РОБОТИ: Контейнеризація додатку.

ЦІЛЬ РОБОТИ: Розгорнути додаток, який складається із кількох контейнерів. Написати bash-скрипт для розгортання додатку

ПОРЯДОК ВИКОНАННЯ РОБОТИ:

Зміст завдання:

1. Ознайомитися із документацією Docker (або іншого контейнеризатора за бажанням студента)
2. Ознайомитися із можливостями Docker compose для налагодження мультиконтейнерного середовища
3. Написати bash-скрипт для розгортання та запуску додатку.

Зміст звіту

1. Описати клієнт-серверний додаток (призначення, функції, варіанти використання)
2. Опис послідовності контейнерування та запуску додатку локально (із скріншотами та наведеними командами запуску).
3. Опис послідовності розгортання та запуску додатку в мультиконтейнерному середовищі (зі скріншотами та командами)
4. Навести лістинг Docker — файлів та опис використаних в ньому функцій
5. Навести лістинг Docker compose — файлів та опис використаних в ньому функцій.
6. Навести лістинг всіх сформованих build-скриптів, що були використані для конфігурації додатку (bash, maven/gradle, yaml, тощо)

Вимоги щодо представлення результатів виконання завдання

1. Звіт має бути створений в електронному вигляді та завантажений на DL. Формат: окремий файл (у форматі *.PDF). Назва звіту: «ПІБ_група_ЛР_1».
2. Тестовий проект, а також bash-скрипт для його розгортання на сервері слід завантажити на GitHub.

ОЦІНЮВАННЯ.

Виконання роботи оцінюється із максимуму 100 балів.

Захист поза терміном карається **“-10”** *балами за кожне ПЗ*

Кожний наступний рівень повинен включати в себе обов’язково виконані в повному обсязі та задокументовані в звіті завдання всіх попередніх рівнів.

ЗАВДАННЯ

Початовий рівень (< 80 балів):

1. Зконфігурувати власний додаток, який може приймати запити через API (наприклад, простий REST API з одним або кількома endpoint'ами) та запустити його локально

2. Створити Docker-контейнер та розмістити в ньому створений додаток

3. Написати bash-скрипт для запуску законтейнеризованого додатку.

Середній рівень (81-89 балів)

4. Налагодити взаємодію додатку, який запущений на Вашому локальному ПК (хості) із додатком, що запущено в контейнері. **Обов'язковим є використання "volume" для організації постійного зберігання даних (ваші дані мають бути доступні, навіть після видалення та повторного створення контейнеру).**

Наприклад (один із варіантів)

- *jar-додаток, що використовує БД, яка запущена в контейнері. При цьому файли БД мають зберігатися у* ***volume.***
- *додаток із функцією збирання та відправки логів на хост. Ваш sidecar (допоміжний) контейнер має збирати та відправляти на хост логи, що були згенеровані в іншому контейнері.*

Високий рівень (90+ балів)

5. Створити docker-compose.yaml скрипт для запуску декількох (2-х або більше) контейнерів, що комунікують між собою за допомогою сутностей docker-compose типу network. При чому, файли БД (або логи та ін.) **обов'язково мають зберігатися у volume.**

Наприклад, запустити в окремих контейнерах під управлянням Docker compose:

1) додаток java (.jar) в контейнері, що побудований на основі java;*

2) сервер БД, до якого звертається основний додаток;

3) менеджер БД (зля зручного управління БД, наприклад pgAdmin, phpMyAdmin тощо, що застосовується для запущеної БД)

6. Розгорнути та запустити за допомогою створеного bash-скрипта додаток.

*** В разі копіювання наданих в прикладах скриптів/додатків/докер-файлів тощо, робота буде оцінена на мінімальний бал (60-66)**