Адміністрування Unix-подібних операційних систем

Практичний проект № 2

НАЗВА РОБОТИ: Використання оркестраторів для автоматизації процесів підготовки проекту до релізу.

ЦІЛЬ РОБОТИ: Навчитися встановлювати та налаштовувати оркестратори. Навчитися адоптувати проект до релізу за допомогою систем оркестрації

ПОРЯДОК ВИКОНАННЯ РОБОТИ:

Етап 1. Підготовка:

1. Ознайомитися із документацією одного з оркестраторів, наприклад, Kubernetis (або будь-якого іншого за бажанням студента, наприклад, Swarm)

2. Розгорнути на локальному комп’ютері довільний контейнеризатор, наприклад, Docker (для цього також можна використати будь-який альтернативний контейнеризатор, наприклад CRI-O або RKT)

3. Розгорнути обраний оркестратор на локальному комп’ютері (для розгортання Kubernetes необхідно встановити Kubernetes CLI: Kubectl (https://kubernetes.io/ru/docs/tasks/tools/install-kubectl/) а також емулятор minikube (https://minikube.sigs.k8s.io/docs/start/)).

Етап 2. Виконання завдання

4. На 60 балів достатньо виконати завдання онлайн на офіційному сайті Kubernetes та сформувати звіт зі скрінами, що підтверджують проходження всіх етапів. В такому випадку, представлені нижче пункти, що включають в себе розгортання оркестратора власноруч, можна пропустити (Подробиці нижче, в розділі “ОЦІНЮВАННЯ”)

5. Зконфігурувати за допомогою обраного контейнеризатора кілька клієнт-серверних додатків в різних образах. (*Дуже важливо! Для налаштування коректної взаємодії Kubernetes та Docker, впевнитися, що доступ до них має група, до якої належить **поточний користувач**, **не тільки root!**).

6. Розгорнути кілька контейнерів, зконфігурованих за допомогою Docker або будь-якого іншого контейнеризатора, наприклад, Containerd CRI-O**,** RKT, на локальному комп’ютері. Задеплоїти контейнери в **один под.** Запустити та перевірити коректність їх роботи **(на оцінку 75 - 89 балів)**.

*7. Розгорнути додатки **в подах** (Pod) за допомогою оркестратора (**на оцінку 90 - 95 балів**).

*8. Налаштувати взаємодію між кількома сервісами, що розгорнуті в різних контейнерах на різних подах (можна на одній ноді) за допомогою оркестратора. **(на оцінку 96+)**

9. Вивести інформацію щодо запущених сервісів: всі запущені поди, об’єм ресурсів, що вони споживають.

Етап 3. Оформлення звіту

10. Описати функції розгорнутих клієнт-серверних додатків (призначення, функції, варіанти використання)

11. Описати послідовність конфігурування та запуску об’єктів (із скріншотами та наведеними командами)

12. Опис послідовності налаштування взаємодії розгорнутих образів за допомогою оркестратора

13. Навести лістинг конфігурацій (подів та інших об’єктів, що були створені)

Етап 4. Результи роботи

14. Звіт має бути представлено в електронному вигляді на DL. Завантажити звіт необхідно окремим файлом (у форматі PDF). Назва звіту: «ПІБ_група_ЛР_3».

15. Тестовий проект, а також всі скрипти конфігурації, що були створені в ході виконання роботи, слід завантажити на GitHub (відповідне посилання необхідно прикріпити в якості коментаря) або завантажити у вигляді архіву (якщо вихідний проект має розмір менше 5 МБ), або на гугл-диск (в такому разі слід прикріпити посилання в якості коментаря на DL).

ОЦІНЮВАННЯ.

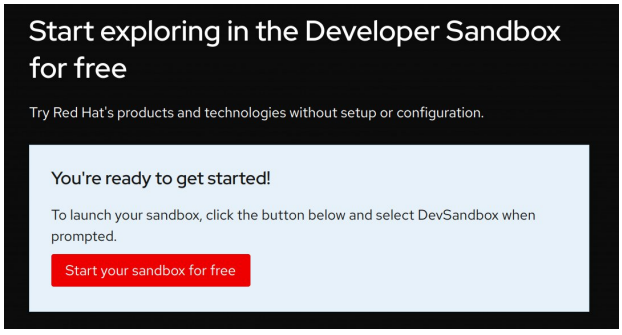
Виконання роботи оцінюється із максимуму 100 балів.

Захист поза терміном карається **-15** *балами за кожне прострочене заняття.*

Вказівки щодо виконання роботи:

На 60 балів необхідно виконати завдання онлайн, що розміщене на офіційному сайті Redhat OpenShift (2 частини)

Частина 1. Зареєструватися на сайті: https://developers.redhat.com/developer-sandbox?extIdCarryOver=true&sc_cid=701f2000001Css5AAC -> виконати онлайн-тьютори *(все відповідне ПЗ у доступі на сайті і не вимагає жодної установки).*



Частина 2. Тьюторіал з налаштування мікросервісу: https://developers.redhat.com/developer-sandbox/activities/learn-kubernetes-using-red-hat-developer-sandbox-openshift

Для отримання максимальної оцінки слід виконати всі запропоновані пункти роботи, представлені нижче. **Кожний наступний рівень повинен включати в себе обов'язково виконані в повному обсязі та задокументовані в звіті завдання всіх попередніх рівнів** (починаючи із середнього)**.**

Середній рівень (75 - 89 балів):

1. Розгорнути локально менеджер контейнерів (наприклад, Docker) та оркестратор (наприклад, minikube або Kubernetes).

2. Зконфігурувати декілька сервісів та запустити їх локально *(можна, використати розгорнутий у docker-compose додаток із попередньої практики).*

3. Імпортувати на кластер за допомогою контейнеризатора додаток, що складається **більше, ніж із одного сервісу**. Достатньо розгорнути сервіси в одному кластері **на одному поді**.

4. Налаштувати обмін даними між контейнерамищо запущені всередині пода за допомогою volum'а типу **emptyDir.** *(об'єктом для обміну даних можуть бути логи, що генеруються в одному контейнері, записуються у volume, далі, збираються іншим контейнером та відображаються)*

5. Вивести інформацію щодо розміщених сервісів: **всі поди; запущені поди**, **запущені контейнери** за допомогою утиліти «kubectl»

Високий рівень (90 - 95 балів) (Включає виконання попереднього завдання)

6. Запустити два або більше контейнерів на різних подах

7. Створити volume типу hostPath, який працює на рівні ноди *(об'єктом для обміну даних можуть бути логи, що генеруються в одному контейнері, записуються у volume, далі, збираються іншим контейнером та відображаються. Іншим об'єктом можуть бути дані, із кешу, записані у відповідний volume та зчитані іншим контейнером із іншого поду).*

Найвищий рівень (96+ балів) (Має включати виконане попереднє завдання):

8. Написати скрипт для запуску всіх налаштувань на оркестраторі

*9. Організувати CI/CD для запуску даного скрипта (на 100 балів)

Інструкції із встановлення необхідгного ПЗ для отримання > **75 балів**

1. Встановити Docker

(подробиці: https://docs.docker.com/get-docker/).

Для Ubuntu:

- *Оновіть список пакетів за допомогою команди*

$ sudo apt-get update

- **Docker → інсталяція***:*

$ sudo apt-get install docker.io

$ sudo systemctl enable docker

$ sudo systemctl status docker

$ sudo systemctl start docker

- Перевірте встановлення (та версію), ввівши docker —version.

$ sudo docker version

- Щоб запустити докер без **sudo**:

$ sudo usermod -aG docker $USER

(перезапустіть оболонку)

$ docker version

2. Встановити kubectl.

(подробиці: https://kubernetes.io/docs/tasks/tools/.)

Він дозволить:

- запускати команди для кластерів Kubernetes
- розгортати програми
- перевіряти та керувати ресурсами кластера
- переглядати журнали

Ви можете знайти більше інформації на сайті:

- **kubectl → інсталяція**

Приклад команд для установки утиліти **kubectl** із бінарного файлу:
$ curl -LO https://storage.googleapis.com/kubernetes-release/release/
$(curl -s https://storage.googleapis.com/kubernetesrelease/release/stable.txt)
/bin/linux/amd64/kubectl
$ sudo install kubectl /usr/local/bin/kubectl
$ kubectl version $ rm kubectl

3. Встановити minikube

(подробиці: https://minikube.sigs.k8s.io/docs/start/)

minikube — це інструмент для створення та керування локальним кластером Kubernetes.

minikube → Системні вимоги
• 2 ЦП або більше
• 2 ГБ оперативної пам'яті
• 20 ГБ вільного дискового простору
• Підключення до інтернету
• Диспетчер контейнерів або віртуальних машин, наприклад: Docker, Hyperkit, Hyper-V, KVM, Parallels, Podman, VirtualBox або VMWare

- **minikube → інсталяція**

Перегляньте примітки щодо встановлення вашої ОС за посиланням:

Приклад установки minikube для Linux із бінарного файлу:
$ curl -LO https://storage.googleapis.com/minikube/releases/latest/minikubelinux-amd64
$ sudo install minikube-linux-amd64 /usr/local/bin/minikube
$ rm minikube-linux-amd64

Перевірте встановлення:
$ minikube version

Приклад результатів виконання команди:
minikube version: v1.20.0
commit: c61663e942ec43b20e8e70839dcca52e44cd85ae