

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Факультет прикладной математики и информатики
Кафедра технологий программирования

Лабораторная работа №4
Методы и средства оркестрации в многоконтейнерных
системах

По дисциплине «Непрерывное интегрирование и сборка
программного обеспечения»

Методические указания по выполнению лабораторной работы

Подготовила:
Давидовская М. И.,
Ст. преподаватель кафедры ТП

Минск, 2024 г.

Содержание

Цель работы.....	3
Задачи работы.....	3
Задания.....	3
Методические указания.....	3
Критерии оценивания.....	3
Содержание отчета.....	3
Задание 1. Подготовка окружения для локального кластера.....	4
Задание 2. Знакомство с ресурсами kubernetes.....	5
Задание 3. Работа с Kubernetes.....	6
Контрольные вопросы.....	6

Цель работы

Изучение создание кластера и работу с kubernetes

Задачи работы

1. Настроить окружение kubernetes.
2. Выполнить задания

Задания

Методические указания

Все результаты лабораторной работы должны быть опубликованы в git-репозитории, ссылка на который доступна в курсе «Непрерывное интегрирование и сборка программного обеспечения»

Критерии оценивания

Для групп 11-13 выполнить все 3 задания, для группы 14 — задания 1-2.

Содержание отчета

1. Цель работы.
2. Вариант задания.
3. Код приложений, конфигурационных файлов.
4. Ответы на контрольные вопросы.

Отчет должен быть опубликован в git-репозитории на github. Все результаты лабораторной работы должны быть опубликованы в git-репозитории, ссылка на который доступна в курсе «Непрерывное интегрирование и сборка программного обеспечения».

В файле Readme проекта на github должна быть ссылка на отчёт. Отчет опубликовать во внешнем хранилище или в репозитории в каталоге /docs. Если в лабораторной работе необходимо написать программу/ы, то отчёт должен результаты тестов по каждой программе и ответы на контрольные вопросы.

Пример оформления файла Readme может быть таким:

```
# Overview

Report on LabRabota1.

# Usage

// Заменить <<link>> и <<folder>> на соответствующие ссылки/названия
To check, please, preview report by <<link>> and source files
in <<folder>>.

# Author

Your name and group number.

# Additional Notes

// СКОПИРОВАТЬ И ВСТАВИТЬ ССЫЛКУ НА свой РЕПОЗИТОРИЙ, НАПРИМЕР
https://github.com/maryiad/lab3-task1-gr16-david
```

Каждая лабораторная работа содержит тексты задач и контрольные вопросы, ответы на которые проверяются преподавателем при приёме работы у студента.

Выполнение студентом лабораторной работы и сдача её результатов преподавателю происходит следующим образом:

1. Студент выполняет разработку программ.
2. В ходе разработки студент обязан следовать указаниям к данной задаче (в случае их наличия). Исходные тексты программ следует разрабатывать в соответствии с требованиями к оформлению, приведёнными в приложении.
3. Студент выполняет самостоятельную проверку исходного текста каждой разработанной программы и правильности её работы, а также свои знания по теме лабораторной работы.

Задание 1. Подготовка окружения для локального кластера

Для работы нам понадобятся утилита kubect и локальный кластер kubernetes. Рассматриваем установку кластера в виртуальной машине,

поэтому рекомендуется использовать утилиту `kind`. Можно использовать и отдельный `minikube`. Утилита `kind` требует наличие Docker в системе.

1. Изучить раздел «Подготовка рабочего окружения» из статьи [«Развертывание кластера Kubernetes»](#)
2. Изучить «Главу 16. Introducing Kubernetes» из книги [The Ultimate Docker Container Book 3rd Edition \(2023\)](#)
3. Установить необходимые утилиты для работы с кластерами `kubernetes`.
4. Создать кластер (см. примеры из п. 1 и 2).
5. Продемонстрировать работающий кластер.

Задание 2. Знакомство с ресурсами `kubernetes`

Все объекты в `kubernetes` — это ресурсы. Чтобы получить список ресурсов используется команда **`kubectl get`**. Чтобы посмотреть список всех возможных ресурсов в кластере с их псевдонимами используется команда **`kubectl api-resources`**. Например, команда **`kubectl get namespaces`** или **`kubectl get ns`** выведет список пространств имен, по которым сгруппированы остальные ресурсы.

Следующий важный ресурс в `kubernetes` — **POD**. Pod — единица работы в `kubernetes`, pod создается, запускается на подходящей ноде, работает и завершается. Pod может состоять из нескольких контейнеров (хотя чаще все же из одного), которые гарантировано запускаются вместе на одной ноде.

1. Изучить раздел [«Знакомство с kubernetes»](#) из статьи [«Развертывание кластера Kubernetes»](#)
2. Изучить «Главу 16. Introducing Kubernetes» из книги [The Ultimate Docker Container Book 3rd Edition \(2023\)](#)
3. Изучить список pods в `kubernetes`. Объяснить, зачем нужны эти компоненты и описать в отчете.
4. Создать том и продемонстрировать его монтирование в pod (см. примеры в «Главе 16. Introducing Kubernetes» из книги [The Ultimate Docker Container Book 3rd Edition \(2023\)](#)).
5. Продемонстрировать работу с ReplicaSet, коллекцией pods (см. примеры в «Главе 16. Introducing Kubernetes» из книги [The Ultimate Docker Container Book 3rd Edition \(2023\)](#)).

Задание 3. Работа с Kubernetes

1. Изучить раздел «[Работа с kubernetes](#)» из статьи «[Развертывание кластера Kubernetes](#)»
2. Изучить «Главу 16. Introducing Kubernetes» из книги [The Ultimate Docker Container Book 3rd Edition \(2023\)](#)
3. Изучить «Главу 17. Deploying, Updating, and Securing an Application with Kubernetes» из книги [The Ultimate Docker Container Book 3rd Edition \(2023\)](#)
4. Проиллюстрировать развертывание приложения (deployment) в кластер Kubernetes.

Контрольные вопросы

1. Что такое kubernetes и зачем он нужен?
2. Расскажите про основные компоненты kubernetes и покажите их в kube-systems
3. Расскажите про создание пода из yaml, основные поля.
4. Расскажите про сервисы ClusterIp и Headless
5. Расскажите про сервисы NodePort и LoadBalancer
6. Расскажите про Ingress
7. Расскажите про Deployment, StatefulSet и DaemonSet
8. Расскажите про Job и CronJob
9. Расскажите про ConfigMap и Secret
10. Перечислите основные команды kubectl