

БЕЗПЕРЕРВНЕ РОЗГОРТАННЯ ОБЧИСЛЮВАЛЬНИХ СИСТЕМ

ПРОГРАМА

навчальної дисципліни підготовки бакалаврів
які навчаються за спеціальністю 126 „Інформаційні системи та технології”
факультету електроніки та комп'ютерних технологій
(шифр за ОПП)

РОЗРОБЛЕНО ТА ВНЕСЕНО:

Львівським національним університетом імені Івана Франка

РОЗРОБНИКИ ПРОГРАМИ:

Флюнт О. Є., канд. фіз.-мат. наук, доцент кафедри радіоелектронних і комп'ютерних систем факультету електроніки та комп'ютерних технологій

Затверджено Вченою радою факультету електроніки та комп'ютерних технологій

“ _____ ” _____ 2020 року, протокол № _____

Голова Вченої ради
факультету електроніки та комп'ютерних технологій

Фургала Ю. М.

Обговорено та рекомендовано до затвердження Навчально-методичною радою факультету електроніки та комп'ютерних технологій

“ _____ ” _____ 2020 року, протокол № _____

Голова Навчально-методичної ради
факультету електроніки та комп'ютерних технологій

Лучечко А. П.

ВСТУП

Курс „**Безперервне розгортання обчислювальних систем**” є вибірковою дисципліною циклу професійної та практичної підготовки для студентів бакалаврів, які навчаються за спеціальністю 126 - Інформаційні системи та технології.

Метою викладання навчальної дисципліни „Безперервне розгортання обчислювальних систем” є ознайомлення студентів із сучасними практиками контролю версій та циклами розробки і впровадження програмного забезпечення, методами перевірки якості програмних додатків, неперервного постачання та розгортання програмних додатків для оволодіння сучасними підходами та інструментами безперервної розробки та безперервного розгортання.

Для досягнення поставленої у навчальному курсі мети студентам потрібно ознайомитись з системою контролю версій Git, системою автоматизованої перевірки якості Selenium, системою віртуалізації та контейнеризації Docker, системою неперервного розгортання Kubernetes, системою контролю процесів Jenkins і ін. Після проходження навчання за цією програмою студенти повинні вміти реалізувати цикл розгортання простого програмного забезпечення від розробки та тестування до автоматизованого розгортання.

Завдання процесу вивчення дисципліни є набуття студентами знань з основних практик процесів розробки, перевірки, інтеграції і розгортання, формування початкових умінь та навиків:

- створення образів програмних додатків з організацією контролю версій та розміщення в репозиторіях;
- здійснювати віртуалізацію програмних процесів за допомогою контейнерів Docker;
- запускати та налаштовувати обчислювальні кластери та керувати їхньою роботою;
- аналізу результатів тестування програмного забезпечення.

Курс повинен забезпечити знайомство студента із основами практиками та інструментами циклу програмного забезпечення. Показником оволодіння студентом матеріалу курсу є вміння здійснювати повний цикл розробки простого програмного додатку від розробки до розгортання.

В результаті вивчення даного курсу студент повинен:

знати основні поняття, визначення, процеси та підходи до тестування програмного забезпечення, сучасні системи та підходи до контролю версій програмних додатків, методи та практики автоматизації запуску програмних процесів, основи віртуалізації за допомогою контейнерів, основи роботи обчислювальних кластерів.

вміти працювати з системами контролю версій програмного забезпечення, проводити тестування програмного забезпечення різних видів та рівнів, автоматизувати запуск різних програм та писати власні скрипти, створювати образи власних програмних додатків та завантажувати їх на репозиторії, віртуалізувати процеси за допомогою контейнерів, створювати та налаштовувати кластери, керувати їхньою роботою.

Міждисциплінарні зв'язки. Для вивчення дисципліни необхідні знання з таких курсів: „Операційні системи”, „Основи програмування”, „Бази даних та знань дисциплін”, „Основи Web програмування”, достатніх для сприйняття категоріального апарату процесів віртуалізації, організації системи контролю версій, основ функціонування операційних систем, зокрема Linux, використання bash оболонки команд.

1. Програма навчальної дисципліни складається з таких змістових модулів:

Змістовий модуль 1. Вступ до безперервного розгортання обчислювальних систем. Сучасні практики повного циклу програмних додатків. Інструменти безперервної інтеграції / розгортання. Вступ до системи контролю версій Git.

Змістовий модуль 2. Вступ до контейнеризації та Docker. Віртуальні машини і контейнери. Автоматизація CI/CD. Віддалене і локальне керування через Jenkins.

Змістовий модуль 3. Призначення та задачі Kubernetes. Створення та налаштування обчислювального кластера.

2. Інформаційний обсяг навчальної дисципліни

Змістовий модуль 1. Вступ до безперервного розгортання обчислювальних систем. Сучасні практики повного циклу програмних додатків. Таблиця інструментів безпекової інтеграції/ безперервного розгортання.

Вступ до системи контролю версій Git. Коротка історія Git. Основи Git. Командний рядок. Установка Git. Початкові налаштування Git. Отримання допомоги. Основи Git. Створення Git-сховища. Запис змін в репозиторій. Перегляд історії комітів. Операції скасування. Робота з віддаленими репозиторіями. Робота з мітками. Псевдоніми в Git.

Розгалуження в Git. Про розгалуження в двох словах. Основи розгалуження і злиття. Управління гілками. Робота з гілками. Віддалені гілки. Перебазування. Git на сервері. Протоколи. Установка Git на сервер. Генерація відкритого SSH ключа. Налаштовуємо сервер. Git-демон. Розумний HTTP. GitWeb. GitLab. Git-хостинг. Розподілений Git. Розподілений робочий процес. Участь в проекті. Супровід проекту.

GitHub. Налаштування і конфігурація облікового запису. Внесення власного вкладу в проекти. Супровід проекту. Управління організацією. Scripting GitHub. Інструменти Git. Вибір ревізії. Інтерактивне індексування. Збереження і очищення. Підпис результатів вашої роботи. Пошук. виправлення історії. Розкриття таємниць reset. Просунуте злиття. Rerege. Виявлення помилок за допомогою Git. Підмодулі. Створення пакетів. Заміна. Сховище облікових даних.

Налаштування Git. Конфігурація Git. Атрибути Git. Хуки в Git. Приклад примусової політики Git. Git і інші системи контролю версій. Git як клієнт. Міграція на Git. Git зсередини. Об'єкти Git. Посилання в Git. Pack-файли. Специфікації посилань. Протоколи передачі даних. Обслуговування сховища та відновлення даних. Змінні оточення.

Змістовий модуль 2. Вступ до контейнеризації та Docker. Порівняння віртуальних машин і контейнерів. Встановлення Docker. Hello, World в Docker. Архітектура Docker. Зображення та контейнери. Реєстри. Повна картина. Маніпулювання контейнерами. Запуск контейнерів. Перерахування контейнерів. Перезапуск контейнерів. Очищення звисаючих контейнерів. Запуск контейнерів в інтерактивному режимі. Створення контейнерів за допомогою виконуваних зображень. Запуск контейнерів у відокремленому режимі.

Виконання команд всередині запущеного контейнера. Запуск оболонки всередині запущеного контейнера. Доступ до журналів із запущеного контейнера. Зупинка або ліквідація запущеного контейнера. Картування портів. Демонстрація ізоляції контейнера. Створення власних зображень. Основи створення зображень. Створення виконуваного зображення. Вміщення експрес-програми. Робота з томами. Багатоетапні збірки. Завантаження вбудованих зображень у Docker Hub. Робота з багатоконтейнерними програмами за допомогою Docker Compose. Основи складання. Послуги лістингу. Виконання команд всередині запущеної служби. Запуск оболонки всередині запущеної служби. Доступ до журналів із запущеної служби. Припинення запуску служб. Складання програми з повним стеком.

Автоматизація CI/CD. Встановлення Jenkins на Linux Ubuntu. Адміністрування Jenkins. Керування плагінами Jenkins. Найпростіші Jobs Jenkins включаючи Deployment. Додавання Slave/Node в Jenkins. Віддалене и локальне керування через Jenkins CLI Client. Deployment з GitHub. Автоматизація запуску Build Job. Jenkins Build Triggers. Автоматизація запуску Build з GitHub. Jenkins trigger from GitHub, Jenkins webhook. Build з параметрами. Deploy в AWS Elastic Beanstalk. Запуск Groovy Script. Обнулення лічильника Jenkins Build. Основи Jenkins Pipeline і Jenkinsfile.

Змістовий модуль 3. Призначення та задачі Kubernetes. Призначення інструмента для організації контейнерів. Функції інструментів для оркестрації контейнерів. Базова архітектура: Вузли ведучого-підлеглого, процеси Kubernetes. Основні поняття: поди, контейнери, сервіси. Їхнє призначення. Приклад файлу конфігурації

Вузол і Pod. Служба та вхід. ConfigMap і Secret. Томи. Розгортання та StatefulSet. Основні узагальнені компоненти Kubernetes Робочі вузли. 3 процеси вузлів: виконання контейнера, Kubelet, Kube Proxy. майстер-вузли. Майстер-процеси. Сервер API. Планувальник. Менеджер контролера. etcd. Мозок кластера. Приклад налаштування кластера

Доступність та масштабованість; аварійне відновлення. Порівнянн Kubernetes і AWS. Мінікубе. Kubectl – Встановлення minikube та kubectl. Створення та запуск кластера minikube. Запуск кластера у режимі налагодження. Статус різних компонентів. Створення під розгортання. Шари абстракції CRUD, застосування файлу конфігурації. Особливості команд

Файл конфігурації Kubernetes. Три частини файлу конфігурації Kubernetes. метадані та специфікація. статус. формат конфігураційного файлу. план для стручків (шаблон). підключення служб до розгортань та стручків (мітка, селектор та порт). підключення розгортання до подів. Підключення послуг до розгортання портів.

Оглядова діаграма створених компонентів Kubernetes. Потік запитів браузера. Розгортання MongoDB. Секрет. Внутрішня служба для MongoDB. Розгортання MongoExpress. ConfigMap. Зовнішня служба MongoExpress.

Простір імен. Простори імен за замовчуванням. Призначення просторів імен. Випадки використання. Організація компонентів. Уникайте конфліктів з іншими командами. Спільний доступ до ресурсів. Обмеження доступу та ресурсів. Характеристики просторів імен. Створення компонентів у просторах імен. Зміна активного простору імен

Ingress. Порівняння Зовнішньої служби з Ingress. Приклади файлів конфігурації YAML для зовнішньої служби та Ingress. Внутрішня конфігурація служби для Ingress. Налаштування Ingress у кластері. Контролер Ingress. Середовище, в якому працює ваш кластер (хмарний провайдер або голий метал). Демонстрація: Налаштування Ingress в Minikube. Встановлення контролера Ingress в Minikube. Створення правила введення. Вхідна інформація за замовчуванням. Введення випадків маршрутизації. Кілька шляхів для одного хоста. Кілька шляхів субдомени або домени. Налаштування сертифіката TLS

3. Рекомендована література

1. McQuaid M., Chacon S. Git in practice. – Manning, 2015. https://www.manning.com/books/git-in-practicea_bid=5688bbf4&a_aid=MikeMcQuaid#toc
2. Chacon S. Git internals // Pro Git. – 2009. – P. 223-250. <file:///home/orest/Downloads/peepcode-git.pdf>
3. Farhan Hasin Chowdhury The Docker Handbook <https://www.freecodecamp.org/news/the-docker-handbook/>
4. Jenkins Handbook www.jenkins.io/doc/book/
5. Kubernetes Handbook <https://tanzu.vmware.com/kubernetes-handbook>
6. Kubernetes Documentation <https://kubernetes.io/docs/home/>
7. Chacon S., Straub B. Pro git. – Springer Nature, 2014. – P. 456.
8. Kubernetes Handbook Non-Programmer's Guide to Deploy Applications with Kubernetes [Stephen Fleming](#)
9. Docker Engine overview <https://docs.docker.com/engine/>
10. Jenkins User Handbook jenkinsci-docs@googlegroups.com

4. Форма підсумкового контролю успішності навчання

Підсумкову оцінку якості засвоєння навчальної програми з дисципліни „Безперервне розгортання обчислювальних систем” визначають за сумарними результатами, як виконання та захисту лабораторних робіт цього курсу, так і за результатами написання двох контрольних робіт.

Рівень засвоєння навчального матеріалу дисципліни визначають, використовуючи рейтингову систему оцінювання. Положення про рейтингову систему оцінювання знань розробляють та затверджується на засіданні кафедри з урахуванням особливостей

професійної підготовки та розподілу навчального часу за видами занять. Це положення входить до складу робочої навчальної програми.

5. Засоби діагностики успішності навчання

Оцінка якості засвоєння навчальної програми включає поточний контроль успішності та модульний контроль успішності. Для поточного контролю засвоєння студентами навчального матеріалу передбачається виконання та захист лабораторних робіт, перелік яких наводиться в робочій навчальній програмі. Для модульного контролю засвоєння студентами навчального матеріалу передбачається виконання двох модульних контрольних робіт, порядок проведення та зміст яких наводяться в робочій навчальній програмі. Для організації індивідуальної роботи студентів передбачається написання рефератів, перелік тем яких встановлює робоча навчальна програма.