

Część 1: WPROWADZENIE

WPROWADZENIE DO CI/CD/CD	<p>Wprowadzenie do kursu i projektu Cykl życia oprogramowania Metodyki kaskadowa i zwinne Pojęcia: Continuous Integration, Delivery i Deployment Opis projektu, Wykorzystywane repozytoria GitHub Wprowadzenie do IDE Visual Studio Code Vagrant i maszyna wirtualna</p>
GIT 1	<p>Rodzaje systemów kontroli wersji Konfiguracja i instalacja GITa Budowa GITa - trzy stany, commit Podstawowe komendy w pracy z commitami Wprowadzenie do pracy z gałęziami Repozytorium zdalne, GitHub Wprowadzenie do pracy z repozytorium zdalnym</p>
DOCKER 1	<p>Wprowadzenie do konteneryzacji Instalacja i konfiguracja Dockera Obrazy, a kontenery Podstawowe komendy Sieci w Docker Perystencja danych - wolumeny i bind mounts Wprowadzenie do Dockerfile</p>
PROJEKT CI/CD/CD 1	<p>Utworzenie repozytorium z aplikacją i konfiguracją Jenkinsa Uruchomienie Jenkinsa w kontenerze. Podstawy obsługi Jenkinsa: - ustawienia - joby - bezpieczeństwo - sekrety i poświadczenia (credentials) - instalacja oraz konfiguracja pluginów i narzędzi - widoki - parametryzacja jobów - dodatki (plugins) Instalacja dodatków (pluginów) Jenkinsa wewnątrz Dockerfile</p>



ROADMAPA KURSU DEVOPS CORE

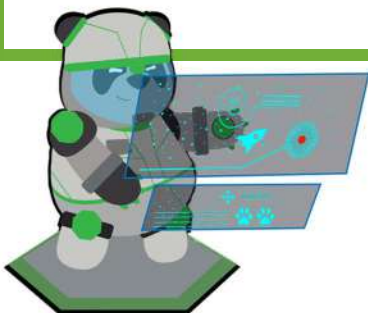
Część 2: CONTINUOUS INTEGRATION

DOCKER 1	Dockerfile - rozwinięcie: <ul style="list-style-type: none">- dostępne instrukcje- ADD VS COPY- ENTRYPOINT VS CMD- .dockerignore Dockeryzacja aplikacji w języku Python (Frontend + Backend) Docker registry Docker-compose: <ul style="list-style-type: none">- podstawowe komendy i dostępne instrukcje- zmienne w docker-compose i środowiskowe- sieci- dziedziczenie
TESTY	Wprowadzenie do testowania Typy testów Poziomy testów Testy manualne a automatyczne Testy Selenium – wprowadzenie Testy selenium – własny test Ciągła analiza jakości kodu - Sonarqube
PROJEKT CI/CD/CD 2	Uruchomienie Jenkinsa i Sonarqube przy użyciu docker-compose Jenkins – konfiguracja jako kod – plugin CasC <ul style="list-style-type: none">- dodawanie haseł i kluczy- konfiguracja URL Jenkinsa- konfiguracja Sonarqube Continuous Integration w Jenkinsie: <ul style="list-style-type: none">- Uruchomienie testów jednostkowych- Uruchomienie analizy Sonarqube
GIT 2	Praca z gałęziami: <ul style="list-style-type: none">- Referencje- Komendy przy pracy z gałęziami- checkout, merge, rebase Zarządzanie zmianami: <ul style="list-style-type: none">- Komendy: git reset, git revert



Część 2: CONTINUOUS INTEGRATION

ARTIFACTORY	Rodzaje repozytoriów w Artifactory Instalacja i konfiguracja Artifactory w Dockerze Konfiguracja Artifactory jako kodu (CasC) Zarządzanie użytkownikami, grupami i dostęпами Konfiguracja repozytorium Docker Registry w Artifactory Automatyzacja dodawania prywatnego rejestru Dockera w Ansible
PROJEKT CI/CD/CD 3	Jenkins agent: - Wprowadzenie - Implementacja przy użyciu konfiguracji jako kodu (CasC plugin) - Dockerfile - Docker in docker Artifactory w docker-compose Budowanie i wysyłanie do Artifactory obrazów Dockera aplikacji przez Jenkinsa
GIT 3	Repozytoria zdalne Podstawowe pliki w repozytorium - Changelog - .gitignore - README - Licencje Etykiety - git tag Cherry pick Zarządzanie cyklem życia aplikacji: - git flow, pull request Zaawansowane komendy (git stash, git blame, git reflog) Automatyzacja przy pomocy hookówGIT w Visual Studio Code
PROJEKT CI/CD/CD 4	Przygotowanie pipeline Continuous Integration w formie Jenkinsfile (Backend i Frontend): - Wprowadzenie - Użycie agenta - Testy jednostkowe - Analiza Sonarqube - Budowanie obrazu kontenera - Publikowanie obrazu do Artifactory



Część 3: INFRASTRUKTURA/ CONTINUOUS DELIVERY

ANSIBLE 1	Terminy Infrastruktura jako Kod, Configuration Management Alternatywy Ansible Podstawowe elementy i pojęcia Instalacja i konfiguracja Ansible Przykładowy playbook Role - wprowadzenie Ansible Galaxy i requirements.yaml
ANSIBLE 2	Role i ich składowe Zmienne Fakty Stworzenie własnych ról
PROJEKT CI/CD/CD 5	Rozwinięcie kodu maszyny wirtualnej - Vagrant + Ansible: - Dodanie roli Ansible - Automatyzacja stawiania kontenerów podczas startu maszyny - Konfiguracja Sonarqube przez REST API
MONITORING	Monitoring – wprowadzenie Alerty i progi Prometheus/Grafana/Alertmanager: - Wprowadzenie - Konfiguracja jako kod Logowanie – ELK Stack (Elasticsearch, Logstash, Kibana): - Wprowadzenie - Grok - Konfiguracja jako kod
PROJEKT CI/CD/CD 6	Uruchomienie Selenium Grid w docker-compose Przygotowanie pipeline Continuous Delivery w formie Jenkinsfile: - Repozytorium stawiającą aplikację poprzez docker-compose.yml - Testy Selenium przy użyciu Selenium Grid Konfiguracja jobów w Jenkinsie jako kodu – Job DSL plugin



ROADMAPA KURSU DEVOPS CORE

Część 3.1: CONTINUOUS DEPLOYMENT – AWS

AWS 1	Wprowadzenie do przetwarzania w chmurze Rodzaje chmur i najpopularniejsi dostawcy Regiony i Availability Zones Zarządzanie budżetem Instancja EC2 Zarządzanie użytkownikami i dostępami (IAM) Konfiguracja Virtual Private Cloud (VPC)
AWS 2	Wprowadzenie do linii komend AWS Simple Storage Service (S3) Elastic Container Registry (ECR) Load Balancer
TERRAFORM 1	Język HCL (Hashicorp Configuration Language) Instalacja i konfiguracja Terraforma Podstawowe komendy Dostawca, zasób (provider, resource) Plik tfstate Provisioner Zmienne Interpolacja wartości Output
TERRAFORM 2	Utworzenie przykładowej infrastruktury wysokiej dostępności wraz z szablonem pliku inventory dla Ansible
PROJEKT CI/CD/CD 7	Instalacja i konfiguracja narzędzi/pluginów – Ansible, Terraform, AWS CLI Przygotowanie pipeline Continuous Deployment na AWS przy pomocy Terraforma i Ansible



Część 3.2: CONTINUOUS DEPLOYMENT – KUBERNETES

KUBERNETES 1	<ul style="list-style-type: none">Wprowadzenie do zarządzania kontenerami w klastrzeBudowa klastraElementy oraz komponenty węzłówZaprojektowanie klastra KubernetesowegoPodstawowe zasoby w klastrze
KUBERNETES 2	<ul style="list-style-type: none">Podstawowe komendyZarządzanie zasobami w klastrzeInfrastruktura jako kodUtworzenie przykładowej aplikacjiZarządzanie aplikacją w klastrze
PROJEKT CI/CD/CD 8	<ul style="list-style-type: none">GitOpsArgoCDPrzygotowanie pipeline Continuous Deployment na Kubernetes przy użyciu ArgoCD

