

# Sprawozdanie

Metodyki DevOps

Ćwiczenia laboratoryjne nr. 6

Pipeline CI/CD

Plan Release Pipeline opracował: Marcin Chrobak

Dla aplikacji:

Node-Chat-App

Data:

24.04.2021

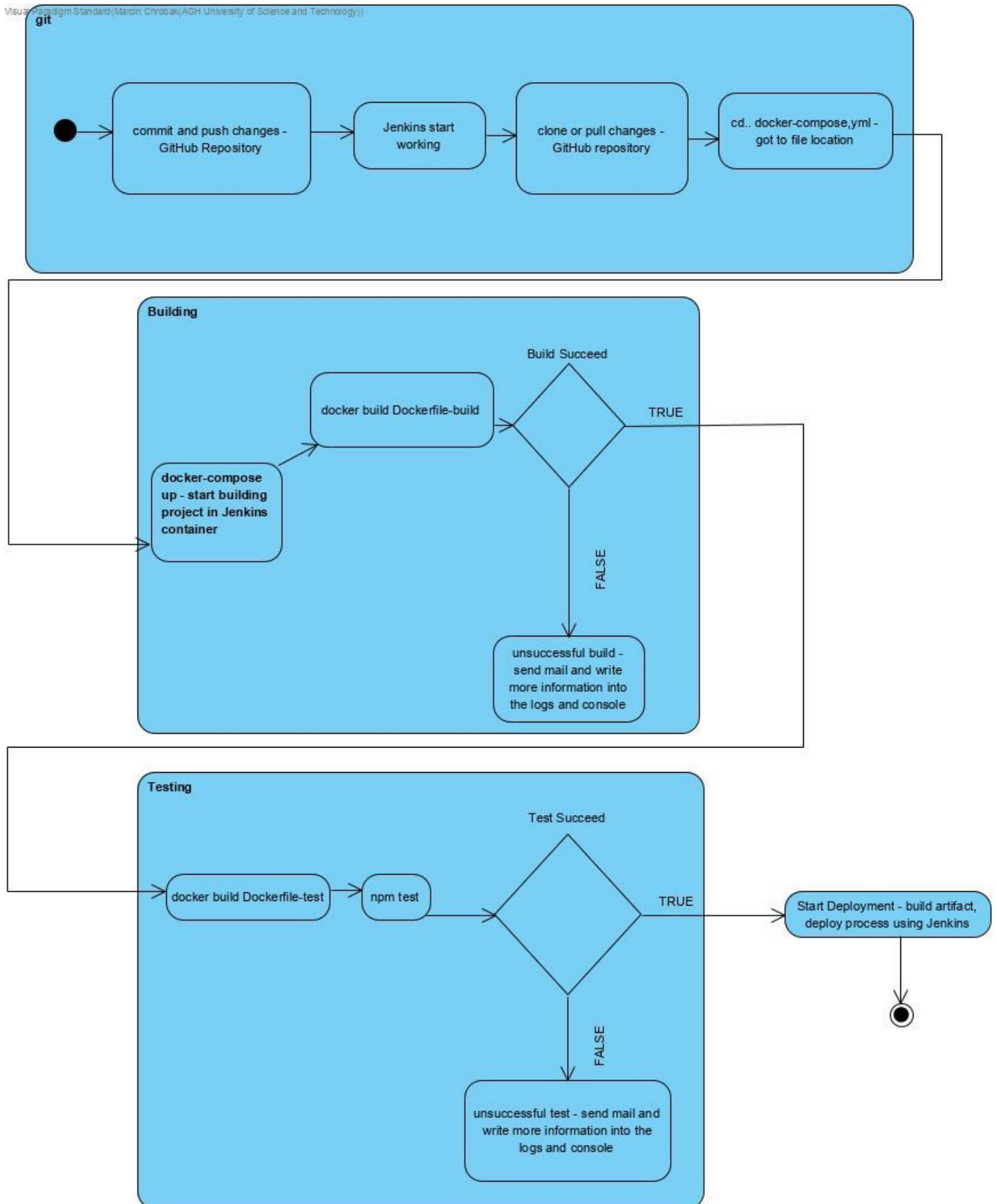
Wykorzystane technologie:

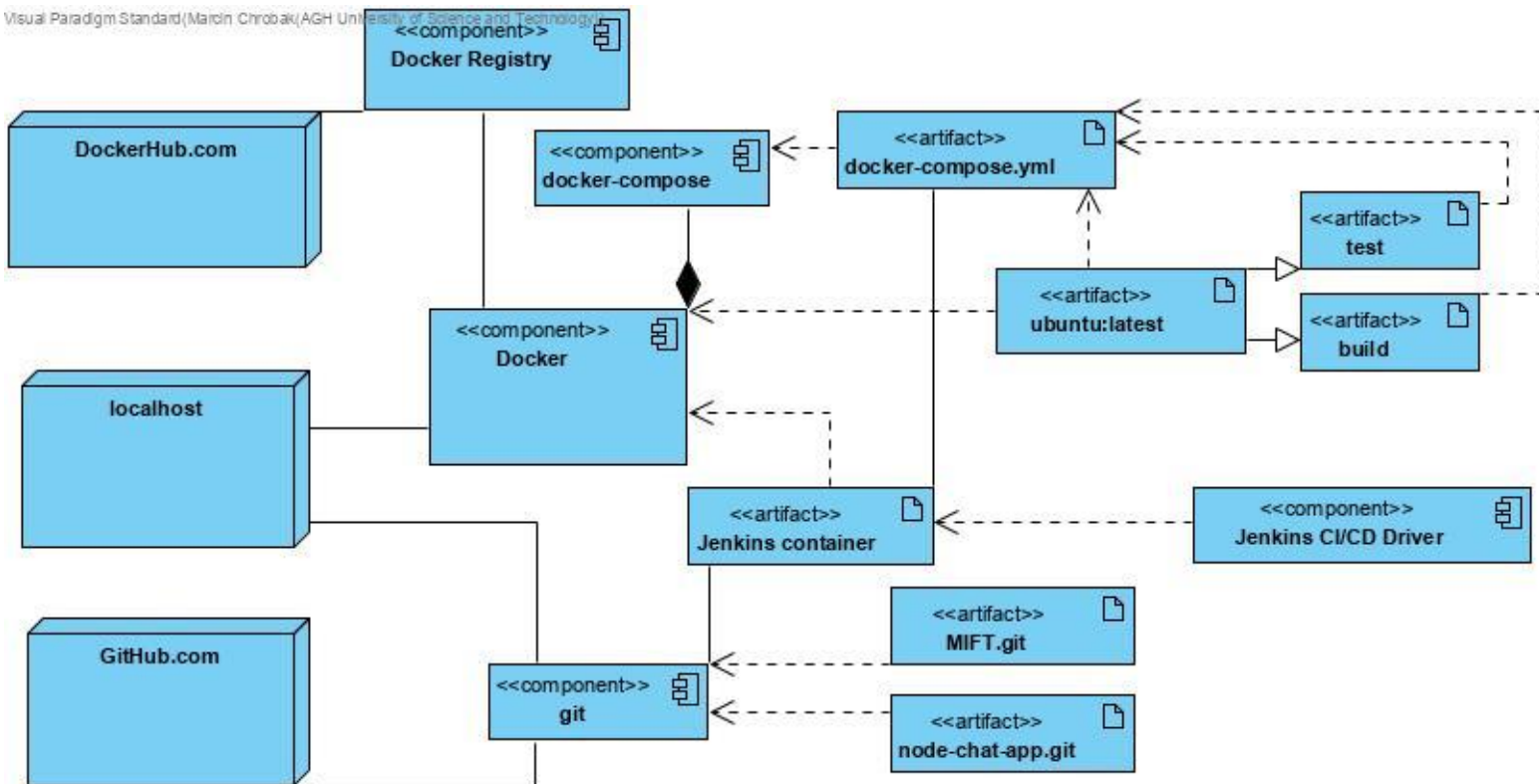
- git: GitHub
- docker
- docker registry: DockerHub
- Jenkins

Diagramy wykonane za pomocą programu Visual Paradigm

# 1. Diagram aktywności - UML Activity diagram

Visual Paradigm Standard (Marion Chrociak\AGH University of Science and Technology)





### 3. Omówienie kroków do wykonania w *CI/CD Pipeline*

- Po dokonania zmian na repozytorium z projektem *Pipeline* jest rozpoczynany, a Jenkins rozpoczyna swoje działanie.
- pobiera dokonane zmiany, przechodzi do odpowiedniego folderu z plikiem *docker-compose.yml* oraz przez wykonanie polecenia *docker-compose up* rozpoczyna budowanie.
- na początku uruchamiany jest plik *Dockerfile-build*, który jest odpowiedzialny za pobranie odpowiednich zależności w celu zbudowania aplikacji.
- jeśli budowanie przebiegło pomyślnie *pipeline* jest kontynuowany, jeśli nie – *pipeline* jest zakończony oraz zostaje wysyłana jest wiadomość z powiadomieniem, a szczegóły są zawarte w logach
- następnym etapem jest testowanie aplikacji, odbywa się to poprzez plik *Dockerfile-test*, który za pomocą *npm test* rozpoczyna testowanie.
- występuje tu podobna sytuacja jak z budowaniem – jeśli zakończy się niepowodzeniem to wysyłana jest wiadomość oraz szczegóły są zawarte w logach.
- jeśli testy są zakończone powodzeniem to rozpoczyna się wdrażanie.