

Release Pipeline Plan

Socket.io-Messenger

Autor: Marcin Kapusta

Technologie:

- Jenkins
- Git
- GitHub
- Docker
- Docker-Compose
- DockerHub

24.04.2021

Pipeline podzielony jest na cztery główne części: inicjację, budowanie, testowanie oraz deployment.

1. Inicjacja:
 - a) Pipeline zaczyna się od merge pull request.
 - b) Uruchamiany jest wówczas job w kontenerze Jenkins. Pobiera on z oddzielnego repozytorium (obecnie jest to MIFT.git) aktualną wersję plików Dockerfile oraz docker-compose.yaml.
 - c) Jenkins uruchamia docker-compose.yaml.
2. Budowanie
 - a) Docker-compose.yaml tworzy dwa kontenery: budowniczego oraz testera. Budowniczy najpierw przygotowuje środowisko do budowania: bazowe środowisko to nodejs alpine, instalowany jest git oraz npm.
 - b) Budowniczy klonuje repozytorium z plikami aplikacji za pomocą „git clone”.
 - c) Budowniczy instaluje aplikację komendą „npm install” wywołaną w katalogu repo aplikacji.
 - d) Wyświetlone zostają informacje o rezultatach budowania. Jeżeli budowanie zakończyło się niepowodzeniem, pipeline jest przerywany.
3. Testowanie:
 - a) Po pomyślnym zbudowaniu aplikacji tworzony jest kontener testera na bazie obrazu budowniczego.
 - b) W katalogu repo aplikacji uruchamiane są testy za pomocą komendy „npm test”.
 - c) Wyświetlenie informacji o testach. Jeżeli kluczowe testy zostały zakończone niepowodzeniem, pipeline jest przerywany. W przeciwnym razie rozpoczynany jest deployment.
4. Deployment:
 - a) Obraz ze zbudowaną aplikacją jest odpowiednio tagowany i wypychany na DockerHub.
 - b) Wyświetlane są informacje o deploymentcie. Jest to koniec pipeline, więc nie ma potrzeby specjalnie go przerywać w razie niepowodzenia.

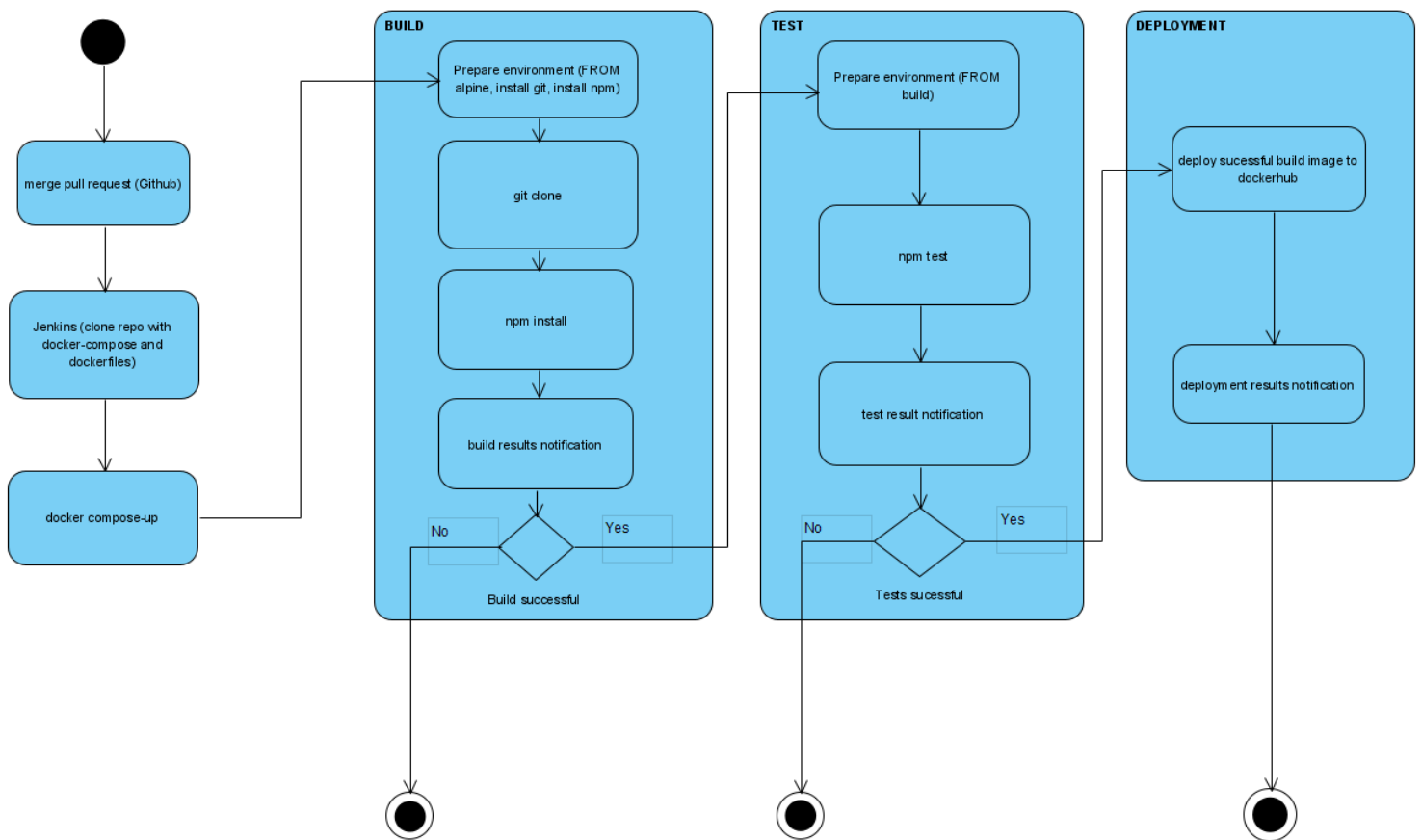


Diagram aktywności

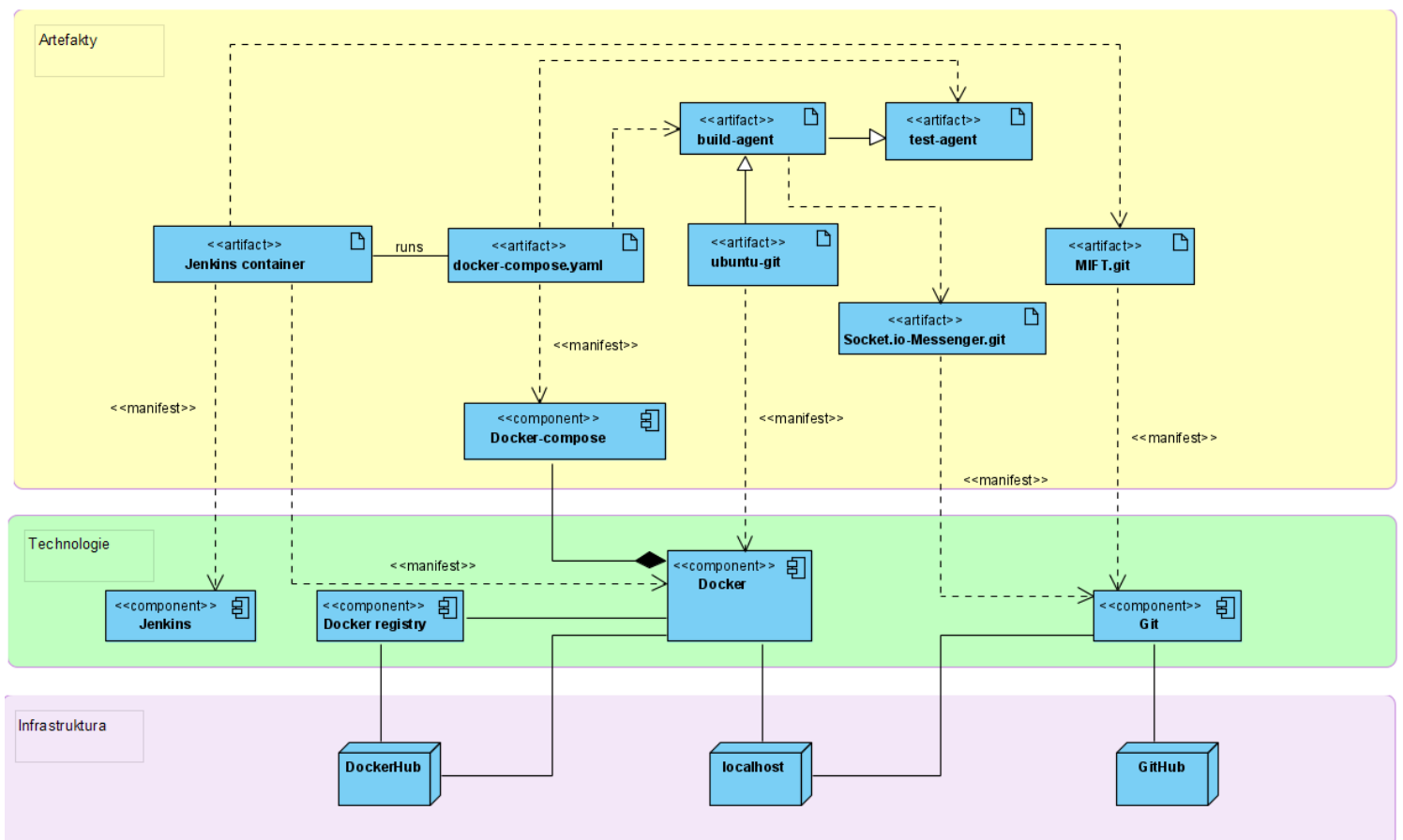


Diagram wdrożeniowy