# Plan wdrożeniowy aplikacji feathers-chat

Plan przygotował Maciej Adamus, 24.04.2021r

#### Wykorzystany stack technologiczny

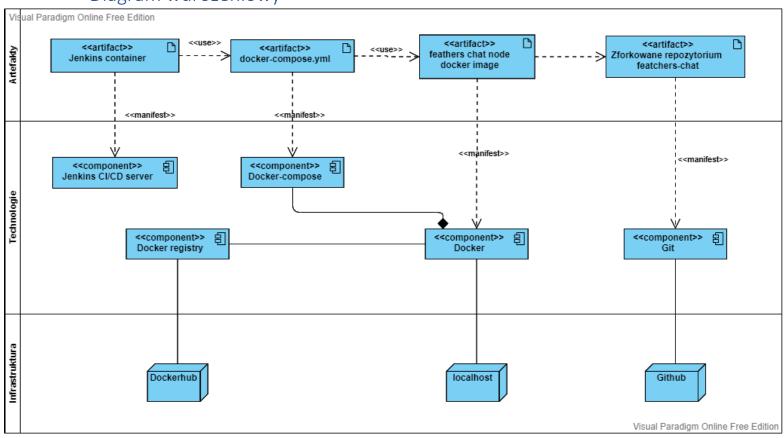
- Docker
- Docker-compose
- Docker registry
- Git
- Jenkins

### Etapy działania CI/CD pipeline

- 1. Pipeline uruchamiany jest po wykryciu nowego commita na gałęzi master forka repozytorium z chatem. Za wykrywanie zmian odpowiedzialny jest jenkinsowy plugin do obsługi gita.
- 2. Zmiany są pobierane do workspace na kontenerze z Jenkinsem przez plugin.
- 3. Uruchamiany jest build dockerowego obrazu z aplikacją, na podstawie dockerfile zawartego w pobranym repozytorium chatu
- 4. Jeżeli podczas buildu wystąpią jakieś błędy, w logach pipeline wypisywany jest odpowiedni komunikat i run zakańczany jest niepowodzeniem
- 5. Na podstawie zbudowanego obrazu uruchamiane są dwa kontenery za pomocą obecnego w repozytorium pliku docker-compose.yml. Jeden z nich uruchamia testy aplikacji za pomocą komendy npm test, drugi uruchamia eslinta za pomocą komendy npm run eslint
- 6. W przypadku wystąpienia błędów w którymkolwiek z uruchomionych kontenerów odpowiedni komunikat zapisywany jest w logach i run zakańcza się niepowodzeniem
- 7. Podsumowanie wyników testów zapisywanie jest w logach
- 8. Obraz zbudowany w punkcie trzecim publikowany jest na dockerhubie

#### Diagram aktywności Visual Paradigm Online Free Edition Build Wypisanie błędu w logach pipeline Build nieudany Zgłoszenie zmiany Pobranie zmian Wypisanie wyniku na podstawie pliku Dockerfile zawartego w repozytorium w repozytorium chatu (commit do mastera) (git plugin do Jenkinsa) Testy Uruchomienie 2 kontenerów za pomocą docker-compose jeden z testami, drugi z eslintem (na podstawie obrazu zbudowanego wczesniej) Deploy Push obrazu na dockerhub Eslint Wypisanie podsumowania nie przeszły zwrócił błedy Wypisanie błędów eslinta w logach nieprzechodzących testów w logach Eslint Testy Wypisanie wyników testów w logach Wypisanie wyników eslinta w logach

## Diagram wdrożeniowy



Visual Paradigm Online Free Edition