

24.04.2021


PIPELINE CI/CD - PLAN

DLA APLIKACJI NODE-CHAT-APP

TECHNOLOGIE:

- Git (w tym github, git hooks)
- Docker (w tym docker-compose)
- DockerHub
- Jenkins (w tym jenkinsfile)

AUTOR: KLAUDIA BARAN
IS ROK 3



PIPELINE CI/CD - PLAN

DIAGRAM AKTYWNOŚCI - OPIS

Poniżej umieszczony diagram przedstawia kolejne kroki wykonywane zainicjowane zgłoszeniem zmian w projekcie, po wdrożeniu go na środowisko produkcyjne.

Pierwszym elementem jest zaistnienie zmian w projekcie sygnalizowane poprzez commit z odpowiednim message (zawartym w skrypcie Git hook), merge i push do repozytorium przy użyciu technologii git'a. Następnie Jenkins pobiera aktualne dane oraz rozpoczyna wdrożenie Pipeline zgodnie z jenkinsfilem. Oznacza to, że w kolejnym kroku dzięki docker-compose oraz zaprojektowanemu wcześniej plikowi compose.yml rozpoczyna budowę dwóch kontenerów, gdzie w pliku (compose.yml) build-agent oparty jest na obrazie docker-build: build (utworzonym z wykorzystaniem Dockerfile-build) oraz test-agent na obrazie docker-test:test (z wykorzystaniem Dockerfile-test). Pierwszy z nich dokonuje konfiguracji, instaluje gita oraz klonuje kod repozytorium komunikatora i instaluje (npm install). W razie niepowodzenia (jest to widoczne w logach) proces zostaje zgłoszony do poprawy i przerwany. Natomiast przy prawidłowej budowie drugi kontener (testujący), zależny od pierwszego (wykorzystuje jego obraz), uruchamia testy (npm test). W razie niepowodzenia proces zostaje zgłoszony do poprawy i przerwany, w przeciwnym wypadku następuje przejście do procesu wdrożenia, utworzenia artefaktów oraz dostarczenia oprogramowania na środowisko produkcyjne.

DIAGRAM AKTYWNOŚCI

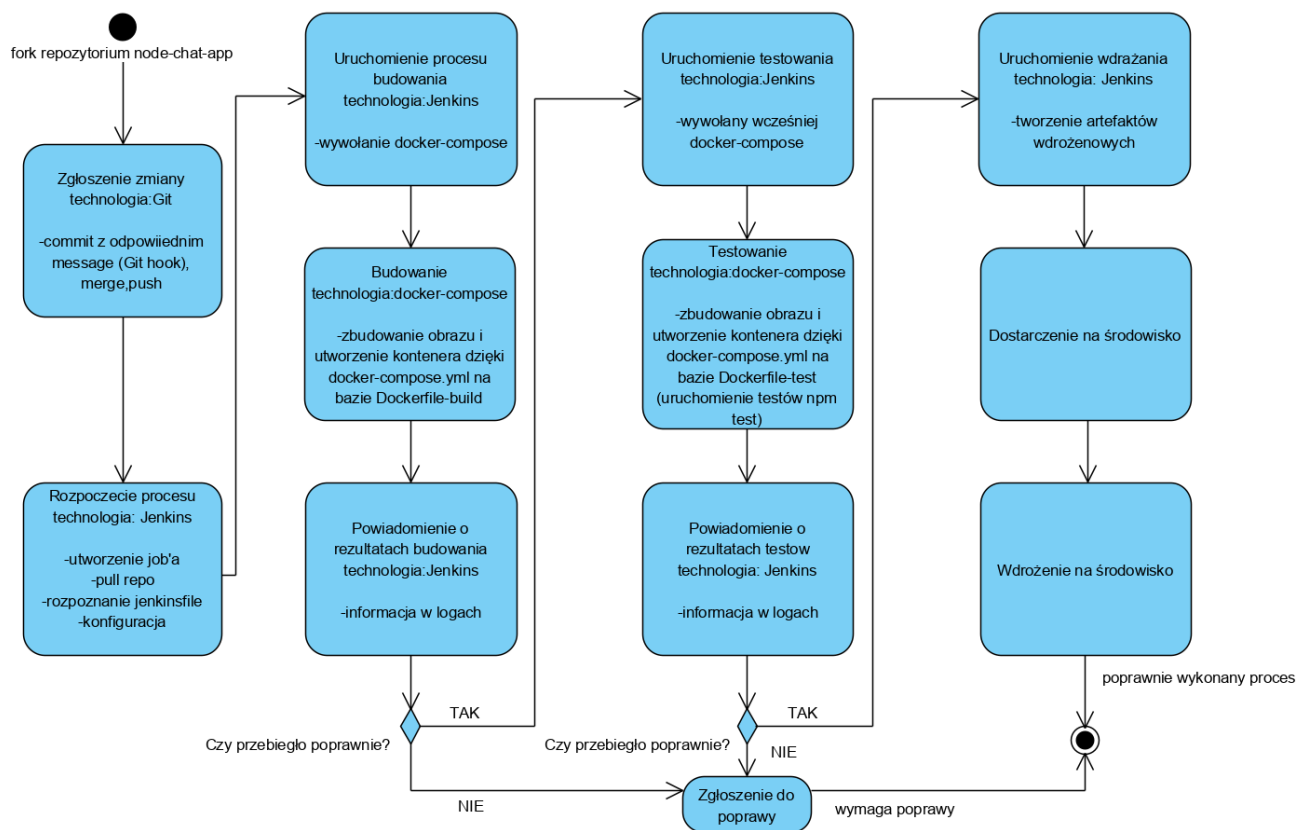


DIAGRAM WDROŻENIOWY

