Plan CI/CD

Pipeline Continuous Integration | Continuous Deployment Jakub Szumilas

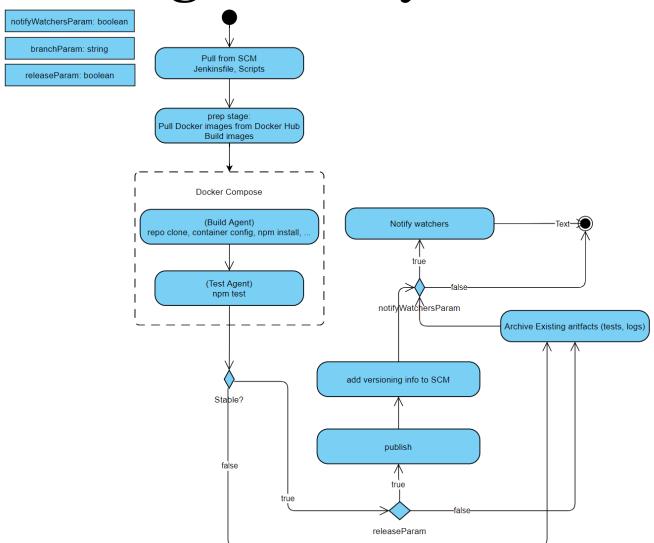
22.04.2021

Wykorzystane technologie

- Docker
 - DockerHub
 - Docker Compose
- Jenkins
 - Groovy
 - Jenkinsfile
- Git
 - GitHub
 - Git Hooks

- Aplikacja: node-chat-app
 - JavaScript ES6
 - NodeJS
 - Express
 - Mocha, Expect
 - Socket.io
 - Mustache.js
 - Moment.js

Diagram aktywności



- Za cały proces odpowiada agent Jenkins, który uruchomiony jest na kontenerze Docker.
 - · Detale działania opisane w skryptowym Jenkinsfile
- Build uruchamiany w ramach pojedynczego pipeline'a podzielonego na stage i stepy.
- Trigger pipeline'u: push na główną gałąź *develop* lub manualnie, z podaniem wskazanej konkretnej gałęzi deweloperskiej.
- Jenkins pobiera potrzebne komponenty:
 - Jenkinsfile
 - Ewentualne dociągnięcie shared libraries Groovy
- Proces buildu i testów wykorzystuje skrypty konfiguracyjne npm
- Uruchamiany jest przez docker-compose
- W zależności od wyniku buildu i testów, przejście do releasu:
 - Release upublicznienie serwisu czatu
 - Nadanie odpowiedniej informacji w repozytorium, oznaczającej stan z danego releasu; wersjonowanie semantyczne
- Zachowanie artefaktów: wyniku testów i logów z pipeline'a/buildu
- Sparametryzowane poinformowanie obserwujących

Diagram wdrożeniowy

