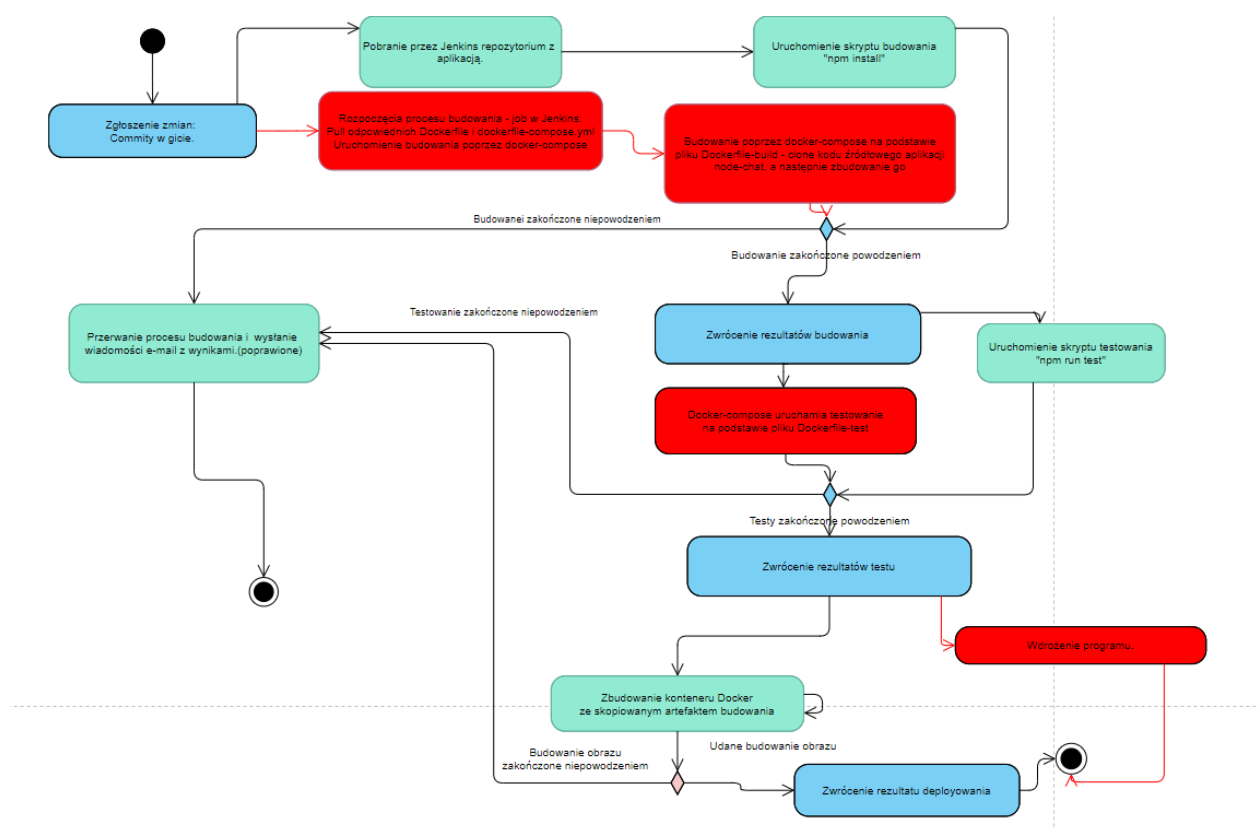


Pipeline CI/CD – Omówienie implementacji dla Node-chat

Wykonał: Wojciech Wróbel Gr.6

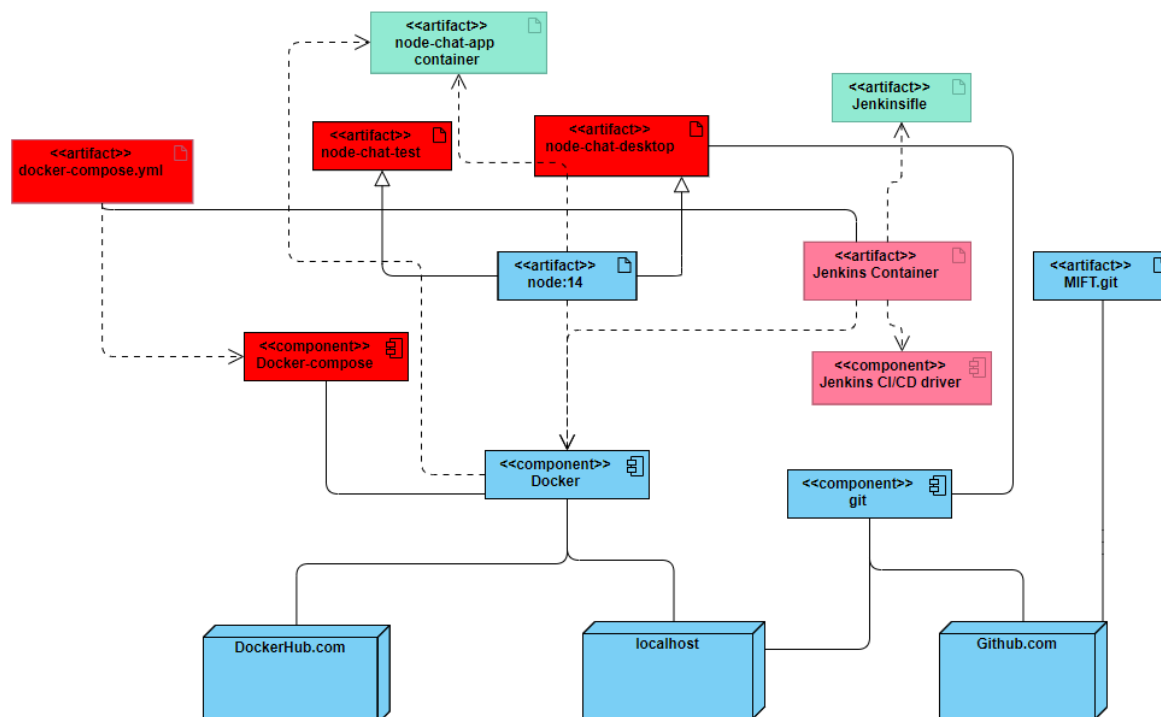
Technologie: Git, Jenkins, Docker

Diagram Aktywności



Krok	Technologia	Plik	Nr lini	Zgodność
Zgłoszenie zmiany w repozytorium	Git, Jenkins	Konfiguracja Jenkins – SCM	-	Tak
Pobranie kodu źródłowego przez Jenkins z repozytorium	Jenkins, Git	Konfiguracja Jenkins	-	W oryginalnym planie zakładane było budowanie poprzez docker-compose.yml
Rozpoczęcie Budowania	Node.js, Jenkins	Jenkinsfile	7-28	W oryginalnym planie zakładane było budowanie poprzez docker-compose.yml
Zwrócenie rezultatów budowania	-	Jenkinsfile	7-28	Tak - dodano wysłanie wyniku testowania na mail
Rozpoczęcia testowania	Node.js, Jenkins	Jenkinsfile	29-49	W oryginalnym planie zakładane było testowanie poprzez docker-compose.yml
Zwrócenie rezultatów testu	-	Jenkinsfile	29-49	Tak – dodano wysłanie wyniku testowania na mail
Rozpoczęcie Procesu Deployowania	Docker, Jenkins	Jenkinsfile Dockerfile	51-73, Cały plik	Dodanie kroku
Zwrócenie rezultatu deployowania	-	Jenkinsfile	51-73	Dodanie kroku
W przypadku niepowodzenia budowania, testowania lub deployowania – wysłanie maila z wynikami niepowodzenia	Jenkins	Jenkinsfile	Sekcje post {...}	Dodanie warunku o wysłaniu maila o niepowodzeniu

Diagram Wdrożeniowy



Nazwa artefaktu	Technologia	Plik	Nr lini	Zgodność
Jenkins Container	Jenkins, Docker	Dockerfile-jenkins	-	Tak
Jenkinsfile	Jenkins	Jenkinsfile	-	Dodano pipeline w Jenkinsfile
docker-compose.yml	Docker	docker-compose.yml	-	Nie skorzystano z docker-compose przy budowanie i wdrażaniu aplikacji
node-chat-app container	Docker	Dockerfile	Caly plik	Po zbudowaniu i przetestowaniu aplikacji jest ona wdrażana na podany kontener
MIFT.git	Git	Repozytorium Github	-	Tak
node-chat-desktop i node-chat-test	Docker	Dockerfile-build Dockerfile-test	-	Nie zbudowano podanych tutaj kontenerów przy budowaniu i wdrażaniu aplikacji