

Imię i nazwisko: Kinga Baryczka

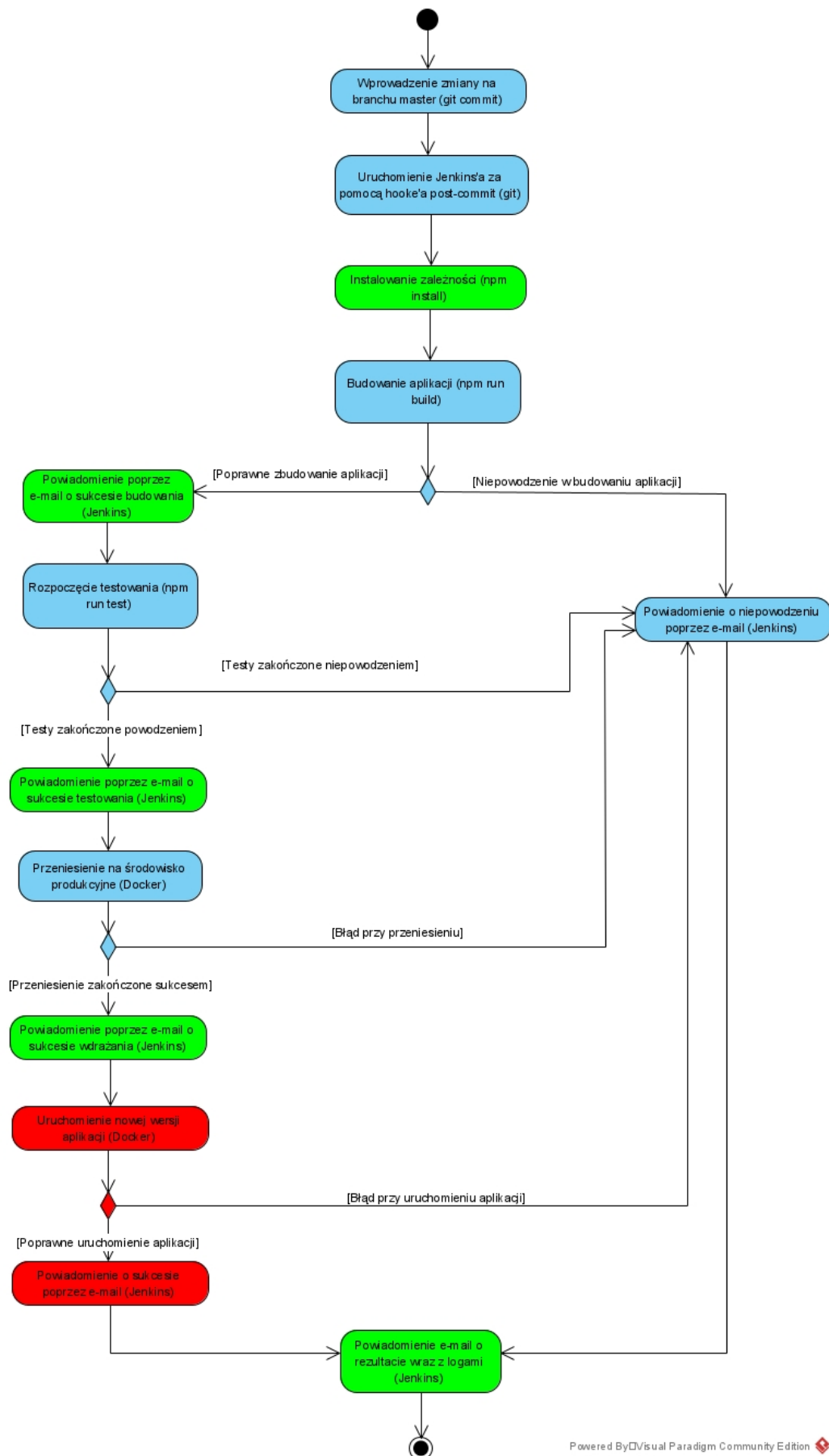
Czas wykonywania: 23.04.2021 r. – 22.05.2021 r.

RZECZYWISTY PLAN WDROŻENIOWY APLIKACJI „DELTACHAT”

ZASTOSOWANE TECHNOLOGIE:

- GIT (W TYM GIT HOOKS),
- DOCKER,
- BASH,
- JENKINS,
- NPM

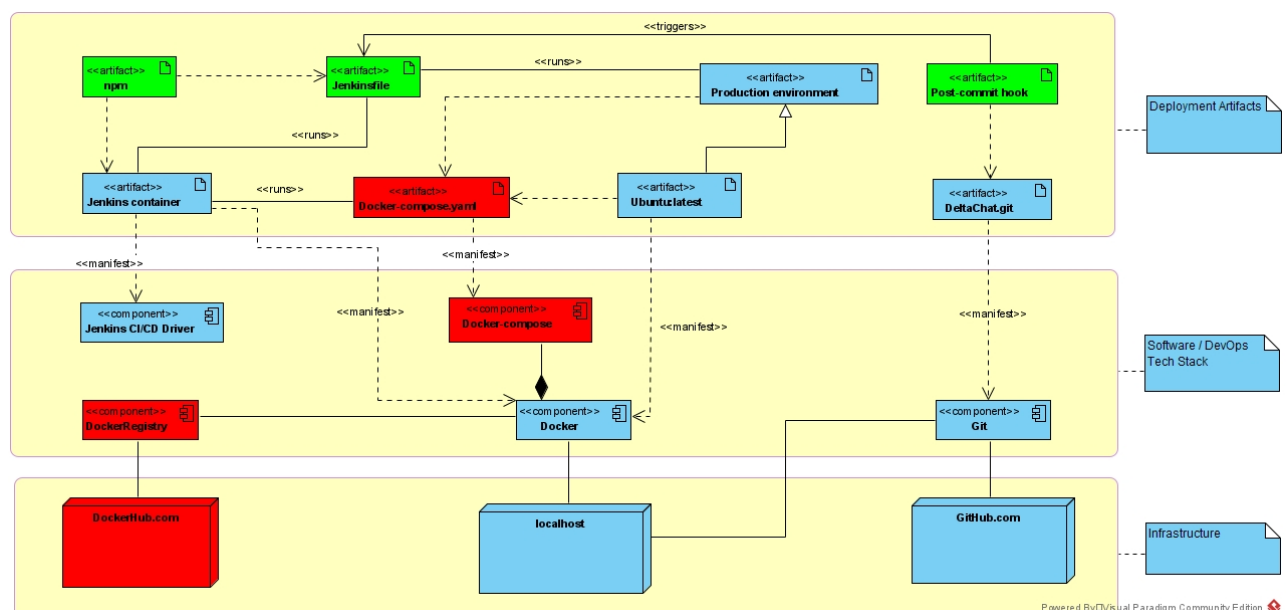
1. Diagram aktywności:



Nazwa kroku	Technologia	Link do pliku	Numer linii	Zgodność z planem, komentarz
Wprowadzenie zmiany na branchu master (git commit)	Git	-	-	Zgodnie z planem – dodanie zmiany poprzez commit na repo Deltachatu
Uruchomienie Jenkins’a za pomocą hooke’a post-commit	Git	Post-commit	4	Zgodnie z planem – post-commit hook uruchamia build w Jenkinsie za pomocą curl
Instalowanie zależności (npm install)	Npm	Jenkinsfile	7	Dodano do planu – Należało zainstalować zależności potrzebne do zbudowania aplikacji
Budowanie aplikacji (npm run build)	Npm	Jenkinsfile	8	Zgodnie z planem – Zbudowanie aplikacji poleceniem npm run build
Powiadomienie o niepowodzeniu budowania poprzez e-mail	Jenkins	Jenkinsfile	11-16	Zgodnie z planem – W razie niepoprawnego zbudowania aplikacji pipeline jest przerywany i wysyłane jest powiadomienie poprzez e-mail
Powiadomienie poprzez e-mail o sukcesie budowania	Jenkins	Jenkinsfile	17-21	Dodano do planu - Wcześniej nie uwzględniono w planie powiadomień o poprawnym zbudowaniu aplikacji
Rozpoczęcie testowania (npm run test)	Npm	Jenkinsfile	27	Zgodnie z planem – Po poprawnym zbudowaniu aplikacji w stage’u test uruchamiano testowanie jednostkowe, integracyjne i e2e za pomocą npm run test
Powiadomienie o niepowodzeniu testowania poprzez e-mail	Jenkins	Jenkinsfile	30-35	Zgodnie z planem – W razie, gdy testy zakończą się niepowodzeniem, pipeline jest przerywany i wysyłane jest powiadomienie poprzez e-mail
Powiadomienie poprzez e-mail o sukcesie testowania	Jenkins	Jenkinsfile	36-40	Dodano do planu – Wcześniej nie uwzględniono w planie powiadomień po poprawnym przetestowaniu aplikacji

Przeniesienie na środowisko produkcyjne (Docker)	Docker	Jenkinsfile Dockerfile.deploy	46	Zgodnie z planem – Przeniesienie aplikacji na odpowiedni obraz dockerowy
Powiadomienie o niepowodzeniu wdrażania poprzez e-mail	Jenkins	Jenkinsfile	49-54	Zgodnie z planem – W razie, gdy wdrożenie zakończy się niepowodzeniem, pipeline jest przerywany i wysyłane jest powiadomienie poprzez e-mail
Powiadomienie poprzez e-mail o sukcesie wdrażania	Jenkins	Jenkinsfile	55-59	Dodano do planu – Wcześniej nie uwzględniono w planie powiadomień po poprawnym wdrożeniu aplikacji
Uruchomienie nowej wersji aplikacji	Docker	-	-	Nie zaimplementowano – Usunięto uruchomienie aplikacji z planu
Powiadomienie o sukcesie uruchamiania aplikacji poprzez e-mail	Jenkins	-	-	Nie zaimplementowano – ponieważ nie uruchamiano aplikacji
Powiadomienie e-mail o sukcesie wraz z logami	Jenkins	Jenkinsfile	63-76	Dodano do planu – powiadomienie o sukcesie bądź niepowodzeniu wraz z załączonymi logami

2. Diagram wdrożeniowy



Etap	Nazwa kroku	Technologia	Link do pliku	Zgodność z planem, komentarz
Infrastructure	GitHub.com	Git	-	Zgodnie z planem – wykorzystano go, aby przechowywać repozytorium z kodem źródłowym
	Localhost	-	-	Zgodnie z planem
	DockerHub.com	DockerHub	-	Usunięto z planu – nie zaimplementowano publikowania obrazów na DockerHub
Software/DevOps Tech Stack	Git	Git	-	Zgodnie z planem
	Docker	Docker	-	Zgodnie z planem
	Docker-compose	Docker-compose	-	Usunięto z planu – zamiast implementacji docker-compose wykorzystano npm install
	Jenkins CI/CD driver	Jenkins	-	Zgodnie z planem – wykorzystano go jako narzędzie do CI/CD
	DockerRegistry	Docker	-	Usunięto z planu – nie publikowano obrazów
Deployment artifacts	DeltaChat.git	Git	Repozytorium	Zgodnie z planem – wykorzystano sforkowane repo z kodem źródłowym aplikacji
	Post-commit hook	Git	Post-commit	Dodano do planu – Post-commit hook za pomocą curl'a uruchamiał pipeline po commicie na repozytorium
	Production environment	Docker	Dockerfile.deploy	Zgodnie z planem – przeniesiono aplikację na środowisko produkcyjne, którym był obraz dockerowy zawierający kod źródłowy aplikacji
	Ubuntu:latest	Docker	-	Zgodnie z planem – wykorzystano obraz ubuntu w środowisku produkcyjnym
	Docker-compose.yaml	Docker	-	Usunięto z planu – nie wykorzystano docker-

				compose, ponieważ był tylko jeden obraz
	Jenkinsfile	Jenkins	Jenkinsfile	Dodano do planu – aby uruchamiać Jenkins pipeline
	Npm	Npm	-	Dodano do planu – aby budować i testować aplikację
	Jenkins container	Jenkins, Docker	-	Zgodnie z planem – wykorzystano go jako kontener dockerowy, w którym uruchamiany był jenkins