

# Rzeczywisty release pipeline dla aplikacji DeltaChat w wersji desktopowej

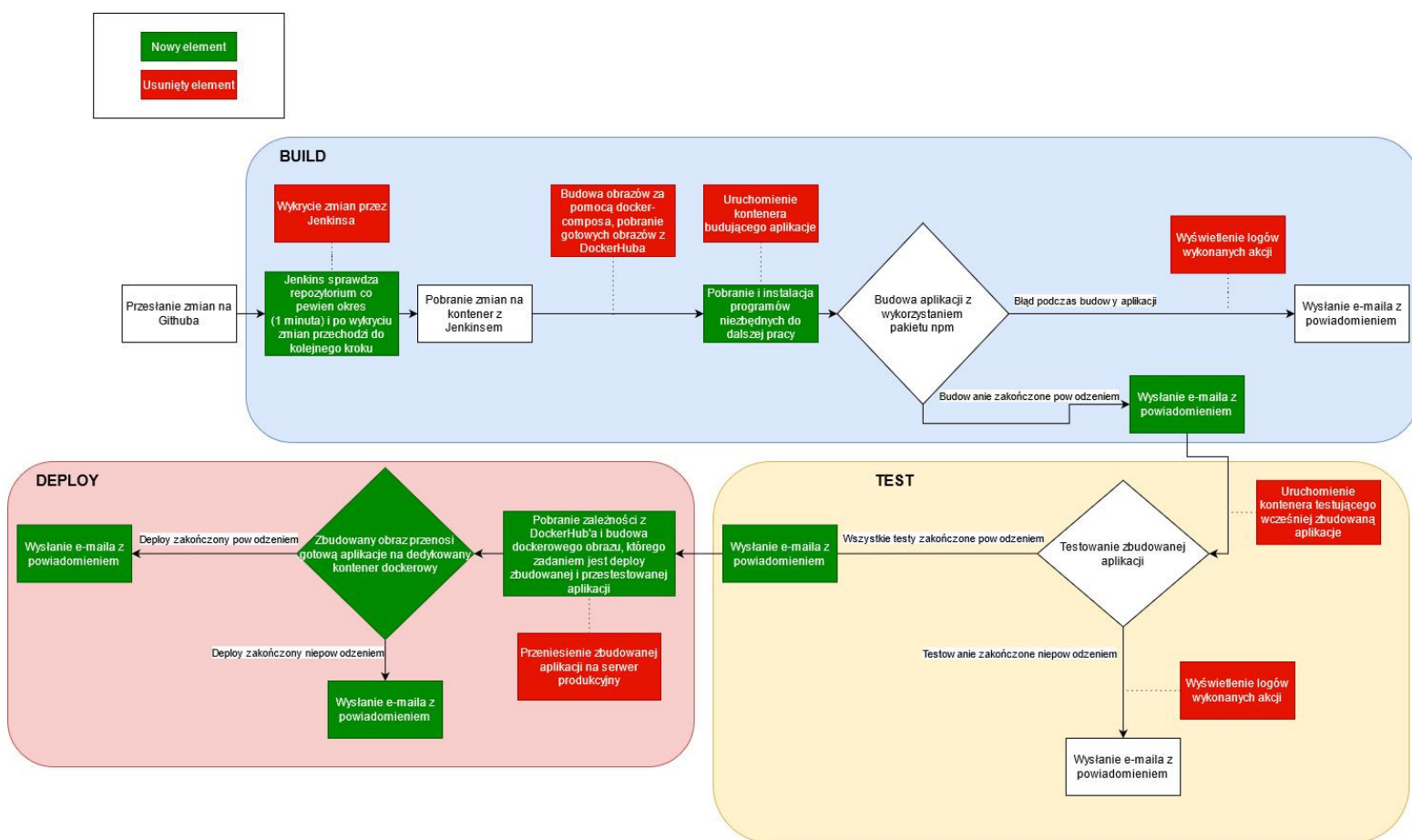
Data wykonania: 20.05.2021

Imię i nazwisko: Krzysztof Klimczyk

## 1. Wykorzystane technologie:

- git
- język programowania: TypeScript/JavaScript
- npm
- node.js
- Docker
- Jenkins

## 2. Diagram aktywności:

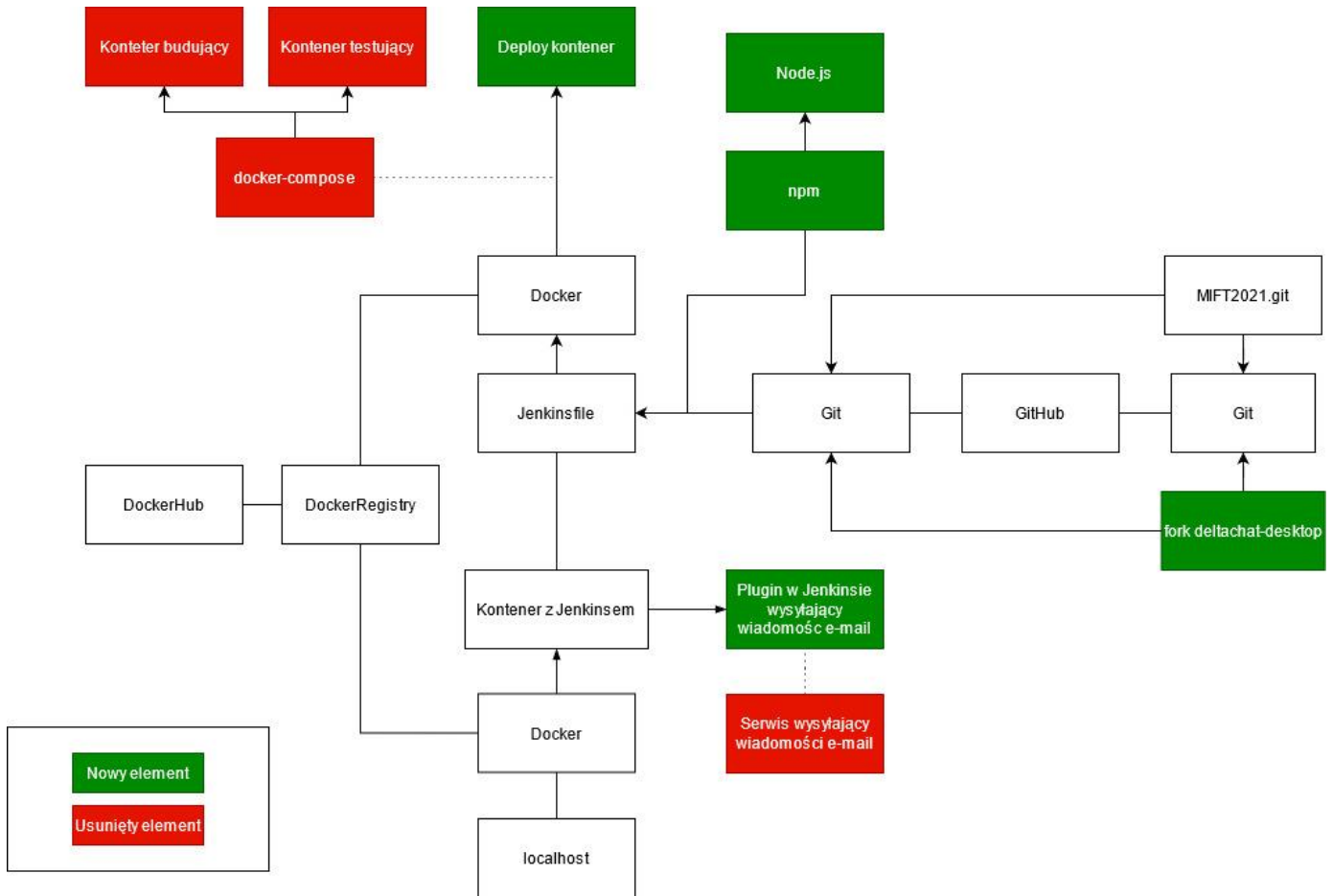


## 3. Tabelaryczne podsumowanie diagramu aktywności:

Nazwa kroku	Technologia	Link	Nr linii	Komentarz
Sprawdzanie zmian na zdalnym repozytorium przez Jenkinsa	Jenkins	<a href="#">GitHub</a>	-	Jenkins sprawdza co pewien okres (1 minuta) czy pojawiły się nowe zmiany na repozytorium. NIEZGODNIE Z PLANEM – Początkowo, to Jenkins miał otrzymywać komunikat z pojawieniem się nowej zmiany na repozytorium ale biorą pod uwagę brak publicznego IP na maszynie z Jenkinsem to rezygnowano z tego kroku.
Pobranie zmian przez Jenkinsa	Jenkins, git	<a href="#">GitHub</a>	9	Jenkins pobiera najnowsze zmiany na lokalne repozytorium. ZGODNIE Z PLANEM

Pobranie i instalacja niezbędnych programów.	Npm, Node.js	<a href="#">GitHub</a>	10-15	Zostają pobrane i zainstalowane najnowsze aktualizacje programów wykorzystywanych do budowy i prawidłowego działania aplikacji. NIEZGODNIE Z PLANEM – Początkowo, budowa aplikacji miała zostać przeprowadzona w dedykowanym kontenerze dockerowym ale z powodu znacznie wydłużonego czasu realizacji tego podejścia zrezygnowano z tego rozwiązania.
Budowa aplikacji	Npm	<a href="#">GitHub</a>	18-19	Budowa aplikacji z wykorzystaniem managera pakietów npm. ZGODNIE Z PLANEM
Wysłanie e-maila z powiadomieniem o błędzie budowania	Jenkins	<a href="#">GitHub</a>	33-39	Jenkins wysyła e-maila z powiadomieniem i załączonymi logami z procesu budowania do określonych użytkowników. NIEZGODNIE Z PLANEM – Początkowo, treść maila miała zawierać tylko powiadomienie ale zmieniono to rozwiązanie aby wysłać dodatkowo logi z danego stage.
Wysłanie e-maila z powiadomieniem o sukcesie budowania	Jenkins	<a href="#">GitHub</a>	24-30	Jenkins wysyła e-maila z powiadomieniem i załączonymi logami z procesu budowania do określonych użytkowników. NIEZGODNIE Z PLANEM – Początkowo, po sukcesie budowania nie była wysyłana żadna wiadomość.
Testowanie zbudowanej aplikacji	npm	<a href="#">GitHub</a>	48	Testowanie aplikacji wykorzystując managera pakietów npm. ZGODNIE Z PLANEM
Wysłanie e-maila z powiadomieniem o błędzie budowania	Jenkins	<a href="#">GitHub</a>	62-68	Jenkins wysyła e-maila z powiadomieniem i załączonymi logami z procesu testowania do określonych użytkowników. NIEZGODNIE Z PLANEM – Początkowo, treść maila miała zawierać tylko powiadomienie ale zmieniono to rozwiązanie aby wysłać dodatkowo logi z danego stage.
Wysłanie e-maila z powiadomieniem o sukcesie budowania	Jenkins	<a href="#">GitHub</a>	53-59	Jenkins wysyła e-maila z powiadomieniem i załączonymi logami z procesu testowania do określonych użytkowników. NIEZGODNIE Z PLANEM – Początkowo, po sukcesie testowania nie była wysyłana żadna wiadomość.
Pobranie odpowiedniego obrazu z DockerHuba'a i budowa obrazu	Docker	<a href="#">GitHub</a>	77	Budowanie obrazu dockerowego za pomocą Dockerfile'a odpowiedzialnego za deploy aplikacji. NIEZGODNIE Z PLANEM – Początkowo, aplikacja miała zostać przeniesiona na osobny serwer ale zastąpiono to rozwiązanie utworzeniem dedykowanego kontenera aby uprościć deploy stage.
Przeniesienie zbudowanej aplikacji na dedykowany kontener	Docker	<a href="#">GitHub</a>	12-13	Przeniesienie gotowej aplikacji na dedykowany kontener. NIEZGODNIE Z PLANEM – Początkowo aplikacja miała zostać przeniesiona na osobny serwer ale zastąpiono to rozwiązanie utworzeniem dedykowanego kontenera aby uprościć deploy stage.
Wysłanie e-maila z powiadomieniem o błędzie budowania	Jenkins	<a href="#">GitHub</a>	90-96	Jenkins wysyła e-maila z powiadomieniem i załączonymi logami z procesu wdrażania do określonych użytkowników. NIEZGODNIE Z PLANEM – Początkowo, po sukcesie wdrażania nie było wysyłane żadne powiadomienie.
Wysłanie e-maila z powiadomieniem o sukcesie budowania	Jenkins	<a href="#">GitHub</a>	81-87	Jenkins wysyła e-maila z powiadomieniem i załączonymi logami z procesu wdrażania do określonych użytkowników. NIEZGODNIE Z PLANEM – Początkowo, po porażce wdrażania nie było wysyłane żadne powiadomienie.

#### 4. Diagram wdrożeniowy:



### 5. Tabelaryczne podsumowanie diagramu wdrożeniowego:

Nazwa artefaktu	Technologia	Link	Nr linii	Komentarz
Kontener z Jenkinsem	Jenkins	<a href="#">GitHub</a>	-	Kontener w którym pracuje Jenkins. ZGODNIE Z PLANEM
Plugin wysyłający wiadomość e-mail	Jenkins	<a href="#">GitHub</a>	-	NIEZGODNIE Z PLANEM – Początkowo, za wysyłanie wiadomości e-mailowych miał odpowiadać osobny serwis. Ten krok został jednak zmieniony tak aby wykorzystywał rozwiązanie wbudowane w Jenkinsa.
Jenkinsfile	Jenkins	<a href="#">GitHub</a>	-	Plik zawierający release pipeline. ZGODNIE Z PLANEM
Deploy kontener	Docker	<a href="#">GitHub</a>	-	NIEZGODNIE Z PLANEM – Początkowo deploy miał odbywać się na osobny serwer jednak zrezygnowano z tego rozwiązania na rzecz dedykowanego kontenera zawierającego zbudowaną aplikację.
MIFT.git	git	-	-	ZGODNIE Z PLANEM
fork deltachat-desktop	git	-	-	NIEZGODNIE Z PLANEM – Początkowo dodatkowe, osobne repozytorium z kodem źródłowym nie było brane pod uwagę.