

<b>Imię i nazwisko:</b> Adrian Dąbrowski	<b>Wydział</b> WIMIIP	<b>Nr ćwiczeń:</b> 8	<b>Grupa lab:</b> 2
<b>Numer indeksu:</b> 401894	<b>Temat:</b> Rzeczywisty plan wdrożenia na przykładzie deltachat- desktop	<b>Data oddania:</b> 07.05.2022	



**AGH**

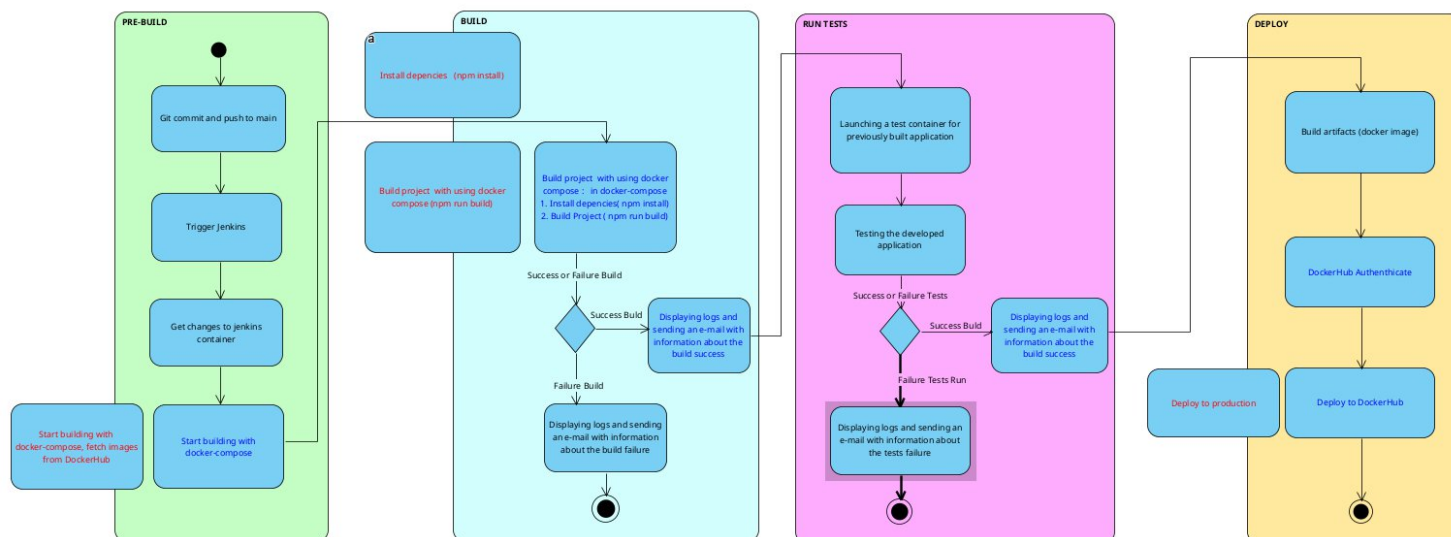
**Rzeczywisty plan wdrożenia na przykładzie  
deltachat-desktop**

# 1. Wykorzystane technologie

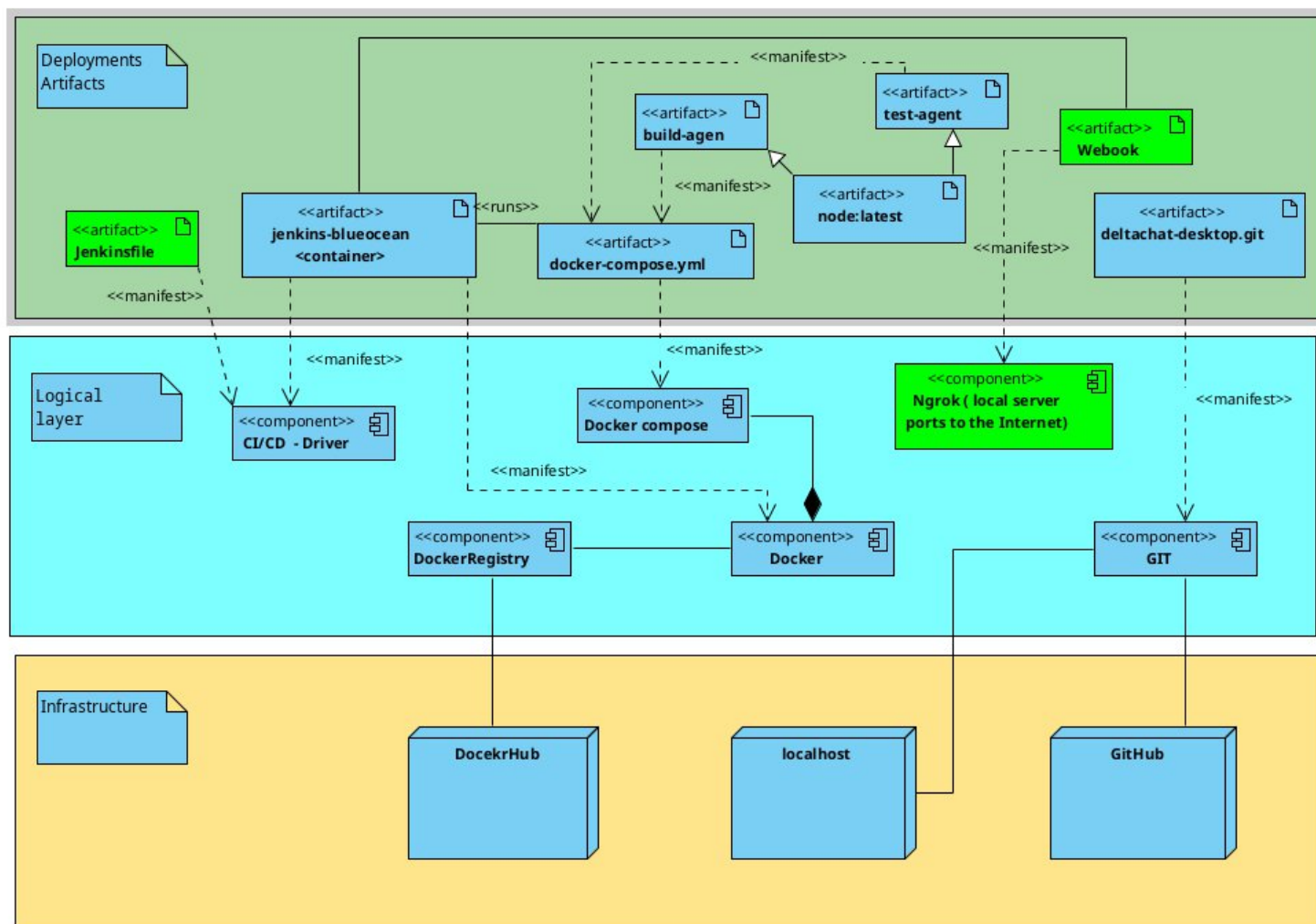
- Git, GitHub
- TypeScript, Nodejs
- Docker, docker-compose,
- DockerHub
- Jenkins
- Repozytorium <https://github.com/deltachat/deltachat-desktop>

Podczas realizacji pipeline wszystkie z powyższych technologii zostały wykorzystane.

## 2. Diagram aktywności



### 3. Diagram wdrożeń



### 4. Tabelaryczne zestawienie zmian (activity)

Krok	Technologia	Odnosnik do pliku	Nr linii	Zgodność z planem
Zgłoszenie zmian na repozytorium	Git/GitHub			Zgodnie z założeniami – dodanie zmian na głównym branch forkowanego repo
Wywołanie pipeline za pomocą Webhooka	Webhook			Zgodnie z założeniami – push na repo powoduje uruchomienie pipeline
Pobranie najnowszej wersji repozytorium	Git			Zgodnie z założeniami – pobranie najnowszej wersji repozytorium
Rozpoczęcie budowania za pomocą docker-compose, obrazy pobierane są z DockerHuba	DockerHub, docker-compose			Częściowo zgodne z założeniami – obrazy nie są pobierane z DockerHuba
Rozpoczęcie instalacji zależności	npm, NodeJS			Niezgodne z pierwotnym planem-punkt był bez sensu, gdyż zależności są instalowane dopiero w docker-compose

Uruchomienie builda za pomocą docker-compose	docker-compose, Jenkinsfile ,Docker	<a href="#">Docker</a> <a href="#">docker-compose</a> <a href="#">Jenkinsfile</a>	- 3-7 7	Zgodnie z założeniami – Build jest tworzony za pomocą docker-compose
Powiadomienie użytkownika za pomocą maila i logów o nieudanej próbie budowania	Jenkinsfile	<a href="#">Jenkinsfile</a>	27-33	Zgodnie z założeniami – użytkownik dostaje email oraz jest informowany o niepowodzeniu w logach
Powiadomienie użytkownika za pomocą maila i logów o udanej próbie budowania	Jenkinsfile	<a href="#">Jenkinsfile</a>	18-24	Niezgodne z pierwotnym planem – użytkownik dostaje email oraz jest informowany o udanym budowaniu aplikacji w logach, wcześniejszy plan nie zakładał takiej opcji
Utworzenie kontenera testującego, bazującego na poprzednio zbudowanym kontenerze (build-agent)	docker-compose, Jenkinsfile ,Docker	<a href="#">Docker</a> <a href="#">docker-compose</a> <a href="#">Jenkinsfile</a>	- 8-14 37	Zgodnie z założeniami – Na podstawie build-agent tworzony jest test-agent
Uruchomienie kontenera oraz testów	NodeJs, npm	<a href="#">Docker</a> <a href="#">Jenkinsfile</a>	3 43	Zgodnie z założeniami – uruchamiane są testy
Powiadomienie użytkownika za pomocą maila i logów o nieprzejęciu testów	Jenkinsfile	<a href="#">Jenkinsfile</a>	57-63	Zgodnie z założeniami – użytkownik dostaje email oraz jest informowany o niepowodzeniu w logach
Powiadomienie użytkownika za pomocą maila i logów o przejęciu testów	Jenkinsfile	<a href="#">Jenkinsfile</a>	48-54	Niezgodne z pierwotnym planem – użytkownik dostaje email oraz jest informowany o udanym przejęciu testów w logach, wcześniejszy plan nie zakładał takiej opcji
Zarchiwizowanie artefaktów	Docker-compose, Jenkinsfile, Docker	<a href="#">Jenkinsfile</a> <a href="#">docker-compose</a>	69 -	Częściowo zgodne z założeniami – image dockerowy jako artefakt, dodatkowo archiwizowany jest artefakt logów z budowania
Autoryzacja DockerHuba	DockerHub, Jenkinsfile	<a href="#">Jenkinsfile</a>	73	Niezgodne z pierwotnym planem – nie przewidywano tego kroku
Deploy na serwer nginx	Jenkinsfile,nginx			Niezgodne z pierwotnym planem – ze względu na aplikacje zmieniono założenia Deployu
Deploy obrazu na DockerHub	DockerHub, Jenkinsfile	<a href="#">Jenkinsfile</a>	73-76	Niezgodne z pierwotnym planem – przewidywano inny sposób podejścia do Deployu