

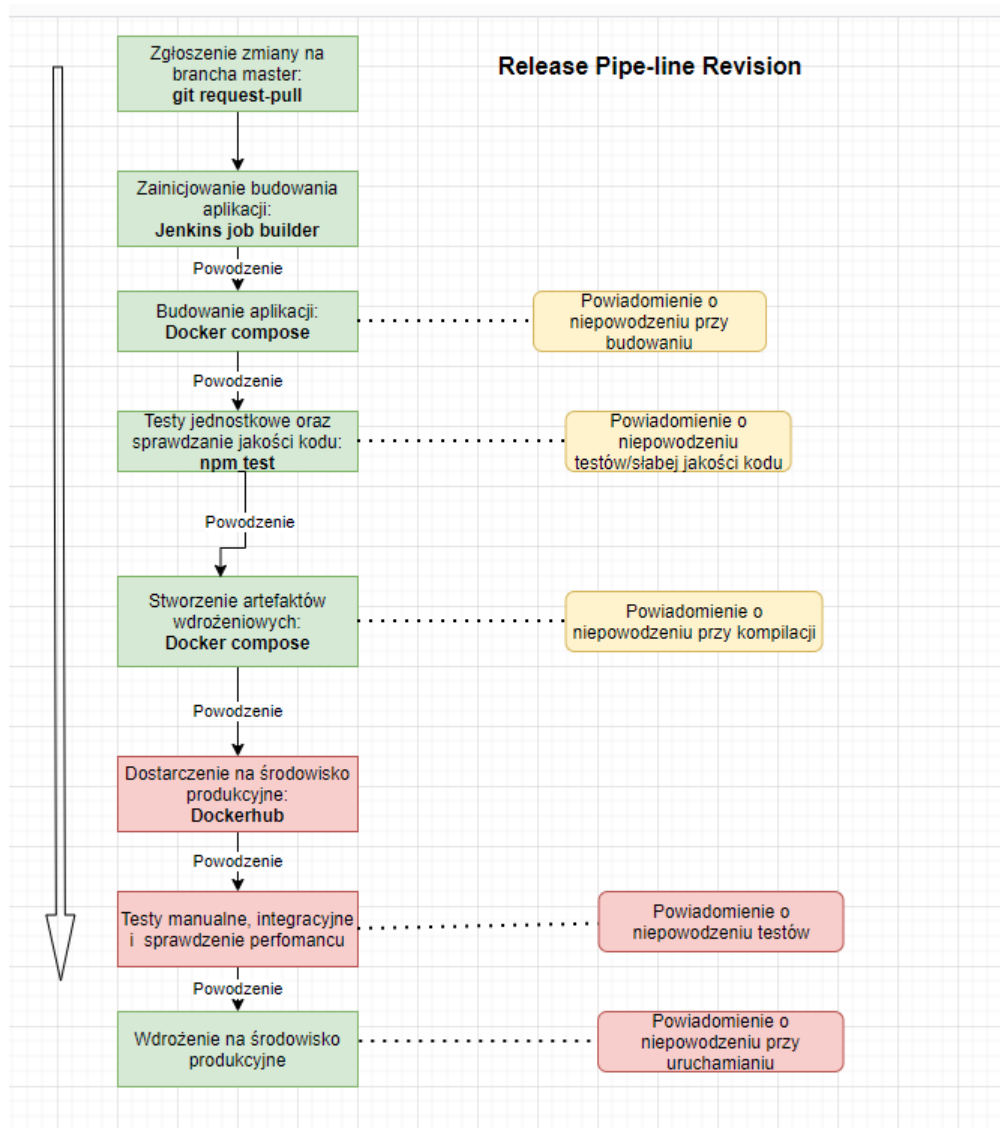
# Plan Release pipeline'a

Przygotowany przez Rafała Widziszewskiego dla aplikacji Deltachat w dniu 29.05.2021r.  
<https://github.com/deltachat/deltachat-desktop>

Planowane technologie wykorzystane w procesie:

- Git
- Jenkins
- Npm
- Docker
- DockerHub

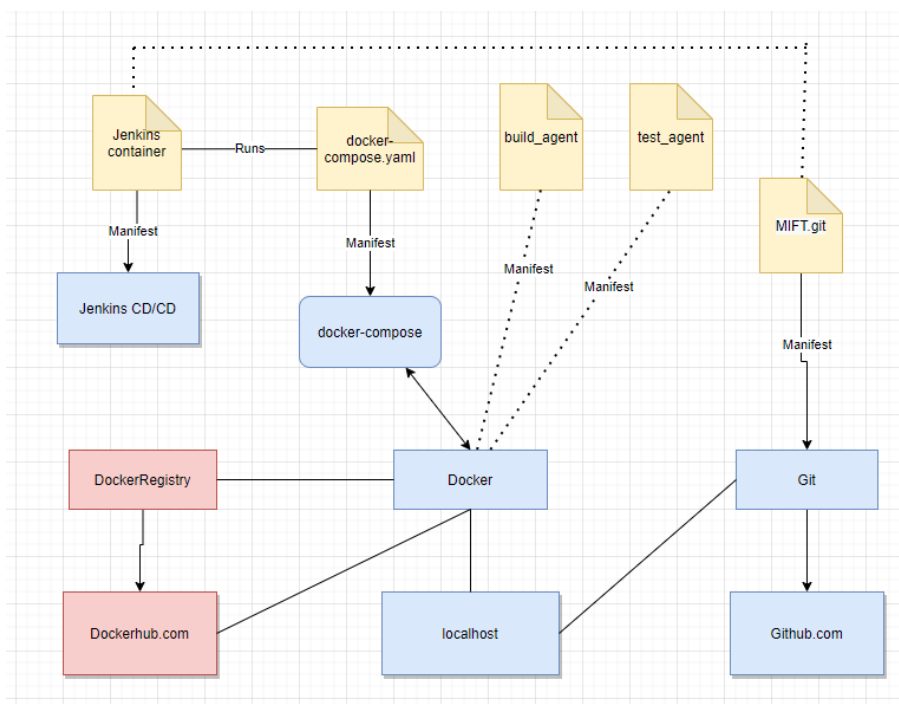
## 1. Diagram aktywności:



## 2. Tabela zestawienie do diagramu aktywności

Nazwa Elementu	Technologia	Zasób	Nr.linii	Zgodność z planem
Git request pull	git	<a href="https://github.com/rwidzisz/deltachat-desktop">https://github.com/rwidzisz/deltachat-desktop</a>	repo	Tak
Jenkins Job Builder	Jenkins	<a href="https://github.com/rwidzisz/deltachat-desktop/blob/master/Jenkinsfile">https://github.com/rwidzisz/deltachat-desktop/blob/master/Jenkinsfile</a>	4-15	Tak
Docker compose	Docker	<a href="https://github.com/InzynieriaOprogramowani-aAGH/MIFT2021/tree/Grupa08-RW306514_Lab08/Grupy/Grupa08/RW306514/Lab08/Docker">https://github.com/InzynieriaOprogramowani-aAGH/MIFT2021/tree/Grupa08-RW306514_Lab08/Grupy/Grupa08/RW306514/Lab08/Docker</a>		Tak, sam proces budowania niepotrzebnie został rozbity na dwa bloki
Powiadomienia	Jenkins	<a href="https://github.com/rwidzisz/deltachat-desktop/blob/master/Jenkinsfile">https://github.com/rwidzisz/deltachat-desktop/blob/master/Jenkinsfile</a>	44-60	Tak
Npm test	npm	<a href="https://github.com/rwidzisz/deltachat-desktop/blob/master/Jenkinsfile">https://github.com/rwidzisz/deltachat-desktop/blob/master/Jenkinsfile</a>	24	Tak
Stworzenie artefaktów wdrożeniowych Docker compose	Jenkins, Docker	<a href="https://github.com/rwidzisz/deltachat-desktop/blob/master/Jenkinsfile">https://github.com/rwidzisz/deltachat-desktop/blob/master/Jenkinsfile</a>	31-43	Tak
Dockerhub	Dockerhub	-	-	Nie
Testy manualne	-	-	-	Nie

### 3.Diagram wdrożeniowy:



### 4.Tabelaryczne zestawienie do diagramu aktywności

Nazwa Elementu	Technologia	Zasób	Nr.liniii	Zgodność z planem
Docker registry	Dockerhub			Nie
Docker	docker	<a href="https://github.com/InzynieriaOprogramowaniaAGH/MIFT2021/tree/Grupa08-RW306514_Lab08/Grupy/Grupa08/RW306514/Lab08/Docker">https://github.com/InzynieriaOprogramowaniaAGH/MIFT2021/tree/Grupa08-RW306514_Lab08/Grupy/Grupa08/RW306514/Lab08/Docker</a>	repo	Tak
Jenkins container	Jenkins	<a href="https://github.com/InzynieriaOprogramowaniaAGH/MIFT2021/tree/Grupa08-RW306514_Lab08/Grupy/Grupa08/RW306514/Lab08/Jenkins">https://github.com/InzynieriaOprogramowaniaAGH/MIFT2021/tree/Grupa08-RW306514_Lab08/Grupy/Grupa08/RW306514/Lab08/Jenkins</a>	repo	Tak
git MIFT	github	<a href="https://github.com/InzynieriaOprogramowaniaAGH/MIFT2021">https://github.com/InzynieriaOprogramowaniaAGH/MIFT2021</a>	repo	Tak
Deltachat github	Github, NodeJS	<a href="https://github.com/rwidzisz/deltachat-desktop">https://github.com/rwidzisz/deltachat-desktop</a>	Repo	Tak

#### Komentarz:

Powiadomienia zostały wykonane tak jak planowałem, w przypadku testów, w przypadku builda oraz deploy nie zostały one zaimplementowane. Nie wyklucza to jednak zaimplementowania tych właściwości w kolejnych wersjach Pipeline’a/Jenkinsfile’a.

Pomimo jednorazowego wrzucenia obrazu na Dockerhub’a nie poczyniłem dalszym kroków, aby był on regularnie używany do przechowywania obrazów. Tak samo jak w przypadku powiadomień, uważam że nie jest to zły pomysł aczkolwiek nie został zaimplementowany z powodu braku czasu.

Jeśli chodzi o testy manualne, jest to sensowny pomysł przy większych projektach i większej grupie osób, która się tymi projektami zajmuje. W naszym Pipeline uważam, że jako osobny krok są zbędne.