

# Omówienie implementacji podstawowego Pipeline z użyciem Docker i Jenkins.

---

*Emil Kobylecki*  
*IS rok III GCL02*  
*27.05.2021*

Omówienie przedstawia:

Pipeline automatycznego testowania i wdrażania nowych wersji rozwijanej aplikacji.

Omówienie zostało opracowane dla:

Aplikacji chatowej open-source Deltachat dostępnej na Github.

<https://github.com/deltachat/deltachat-desktop>

Omówienie zakłada użycie:

Docker, Docker-compose i Docker HUB

Jenkins Pipeline

Git dla Linux oraz Github

Ngrok

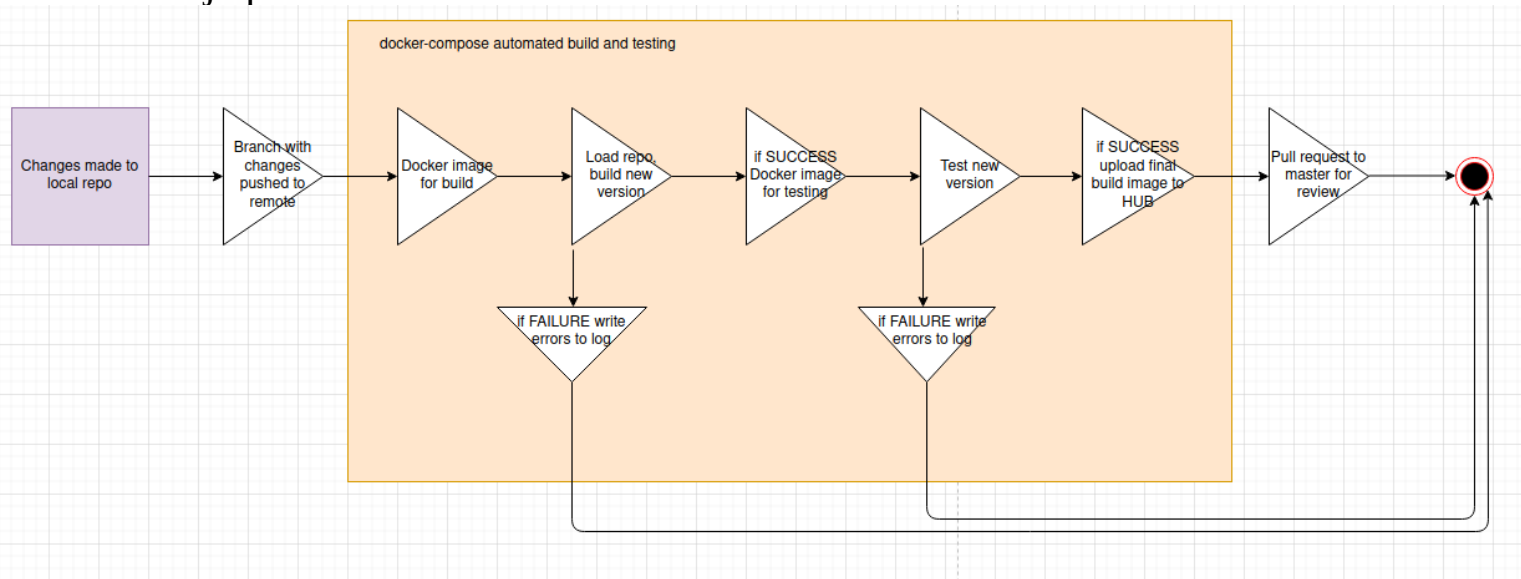
Omówienie zostało opracowane przez:

Emila Kobyłeckiego

w ramach laboratorium nr 10 z przedmiotu DevOps

# 1. Diagram aktywności

Wersja pierwotna:



Wersja obecna:

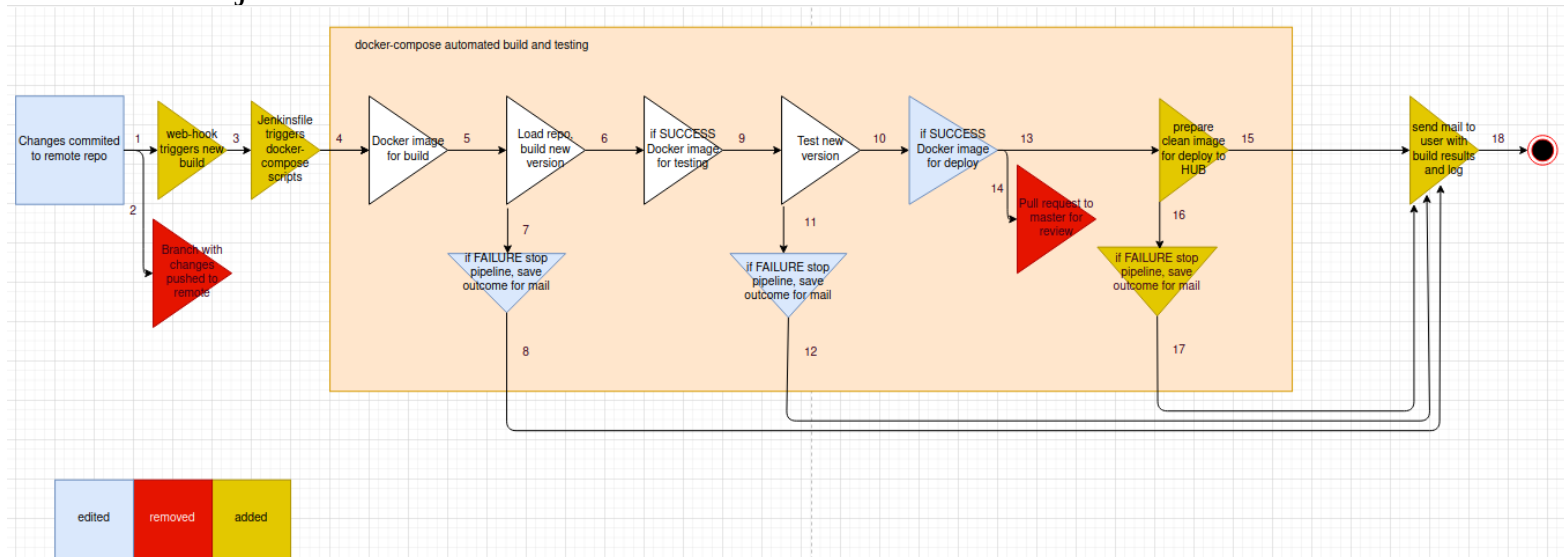


Diagram aktywności uległ zmianie głównie z powodu wprowadzania rozwiązań, które nie były w ogóle założone na początku projektu. Obecny plan Pipeline składa się z następujących kroków:

1. Pipeline rozpoczyna się, kiedy udostępniona zostanie nowa wersja aplikacji w repozytorium (push).
2. Zmiany wykrywane są przy pomocy web-hooka, który następnie przekazuje je na adres usługi Jenkins. Usługa ta działa lokalnie i jest udostępniona pod adresem HTTP przy pomocy programu ngrok.
3. Jenkins działa na bazie Jenkinsfile znajdującego się w repozytorium z aplikacją. Jenkinsfile uruchamia kolejne segmenty pliku docker-compose, który odpowiada za budowanie i testowanie.

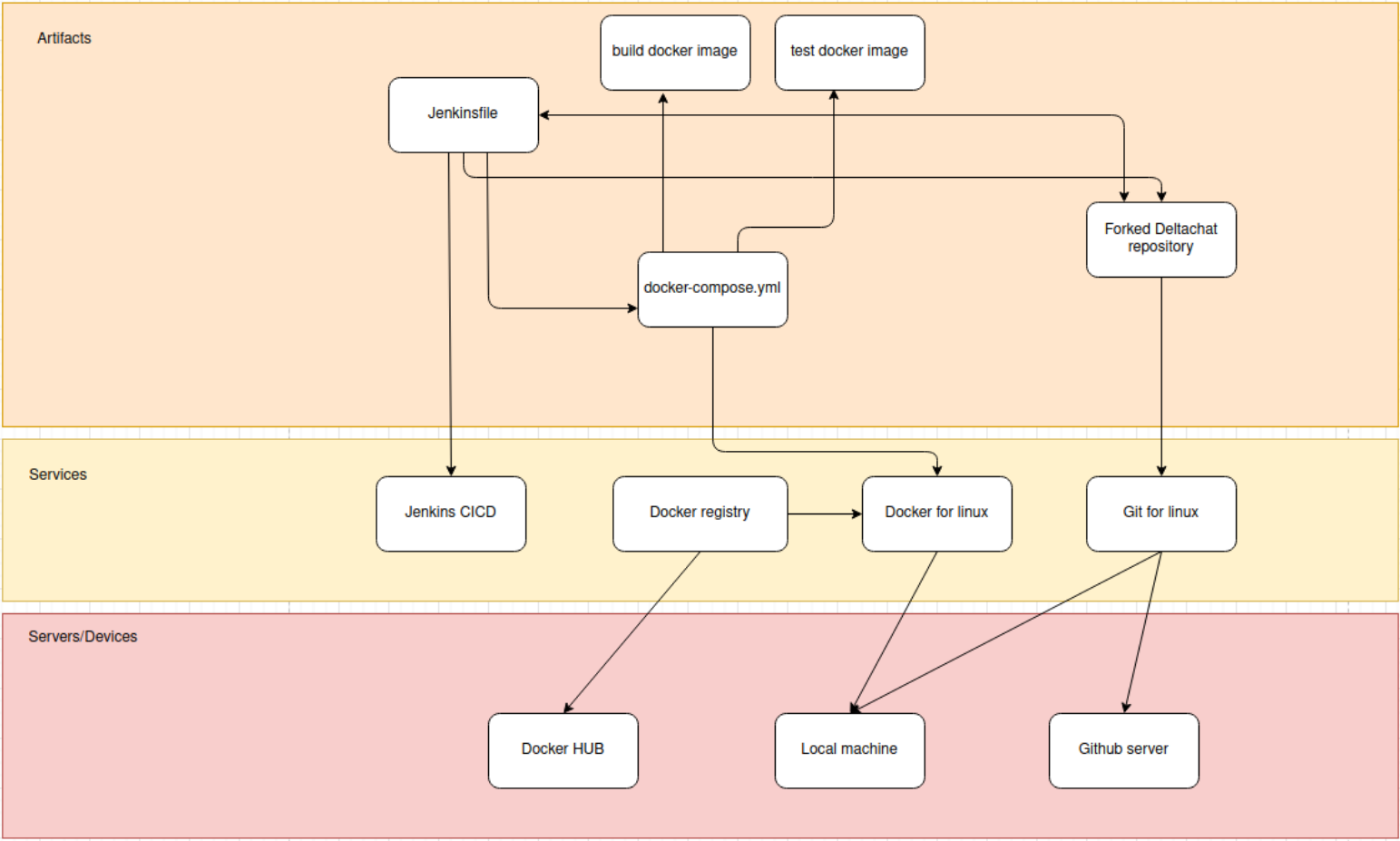
4. Uruchomiony zostaje obraz dockerowy odpowiadający za budowanie aplikacji.
5. Nowa wersja aplikacji jest pobierana i budowana. W przypadku błędu na tym etapie Pipeline jest przerywany, a do użytkownika przesyłana wiadomość e-mail z rezultatami.
6. Uruchomiony zostaje kontener do testowania.
7. Aplikacja jest testowana. W przypadku błędu na tym etapie Pipeline jest przerywany, a do użytkownika przesyłana wiadomość e-mail z rezultatami.
8. Uruchomiony zostaje kontener tworzący wersję gotową do wypuszczenia. Obraz tego kontenera zostaje przygotowany do wysłania na DockerHUB. W przypadku błędu na tym etapie Pipeline jest przerywany, a do użytkownika przesyłana wiadomość e-mail z rezultatami.
9. Do użytkownika zostaje wysłany e-mail z rezultatami każdego z etapów oraz logami Jenkinsa.

Zestawienie tabelaryczne:

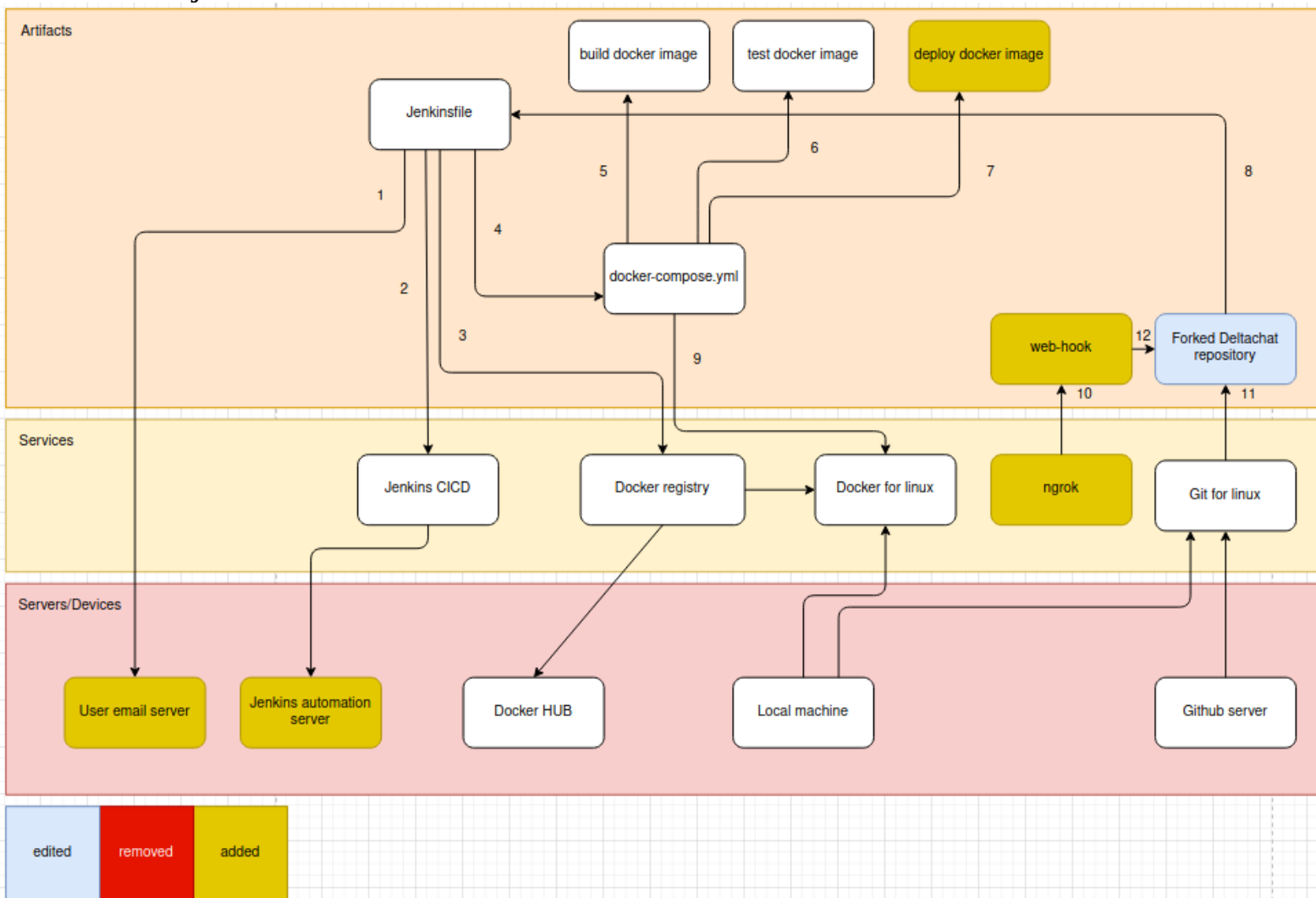
| Nr kroku | Technologia             | Link do pliku   | Nr linii   | Komentarz                                       |
|----------|-------------------------|---|------------|---|
| 1        | Git for Linux           | Konfigurowany przez Github.com                                | 1          | Zmienione: Pipeline zaczyna push                |
| 2        | Ngrok, Jenkins          | Konfigurowane w interfejsie Jenkins                           | 3          | Zmienione: Jenkins udostępniany przez ngrok     |
| 3        | Jenkins, docker-compose | <a href="#">Jenkinsfile</a><br><a href="#">docker-compose</a> | 4          | Zmienione: Jenkinsfile w repo z aplikacją       |
| 4        | Docker, docker-compose  | <a href="#">Obraz</a>   | 5          | Niezmienione                                    |
| 5        | Docker, docker-compose  | jw.   | 6, 7       | Zmienione: e-mail w przypadku niepowodzenia     |
| 6        | Docker, docker-compose  | <a href="#">Obraz</a>   | 9          | Niezmienione                                    |
| 7        | Docker, docker-compose  | jw.   | 10, 11     | Zmienione: e-mail w przypadku niepowodzenia     |
| 8        | Docker, docker-compose  | <a href="#">Obraz</a>   | 13, 15, 16 | Zmienione: całe budowane wersji do wypuszczenia |
| 9        | Jenkins                 | <a href="#">Jenkinsfile</a>                                   | 18         | Zmienione: dodane wysyłania e-maila             |

## 2. Diagram wdrożeniowy

Wersja pierwotna:



Wersja obecna:



Tabelaryczne zestawienie:

| Nazwa artefaktu:            | Technologia:   | Link do pliku:              | Nr linii:     | Komentarz:   |
|-----------------------------|----------------|-----------------------------|---------------|--|
| Jenkinsfile                 | Jenkins        | <a href="#">Jenkinsfile</a> | 1, 2, 3, 4, 8 | Dodane połączenia z usługą e-mail i serwerem Jenkins |
| Build docker image          | Docker         | <a href="#">Obraz</a>       | 5             | -  |
| Test docker image           | Docker         | <a href="#">Obraz</a>       | 6             | -  |
| Deploy docker image         | Docker         | <a href="#">Obraz</a>       | 7             | Dodany   |
| Docker-compose.yml          | Docker-compose | <a href="#">Plik</a>        | 4, 5, 6, 7, 9 | Dodana sekcja dla obrazu Deploy                      |
| Web-hook                    | Git, ngrok     | Konfigurowany przez Github  | 10, 12        | Dodany, łączy Github z Jenkins przy pomocy ngrok     |
| Forked deltachat repository | Git            | <a href="#">Repo</a>        | 8, 11, 12     | -  |