# Metodyki DevOps

| Lab nr: 08 | Pipeline CI/CD dla komunikatora Deltachat Desktop – Aktualizacja | AGH |
|---|---|---|
| Gr.01 | Piotr Apriasz | 402099 |

1. **Wstęp**

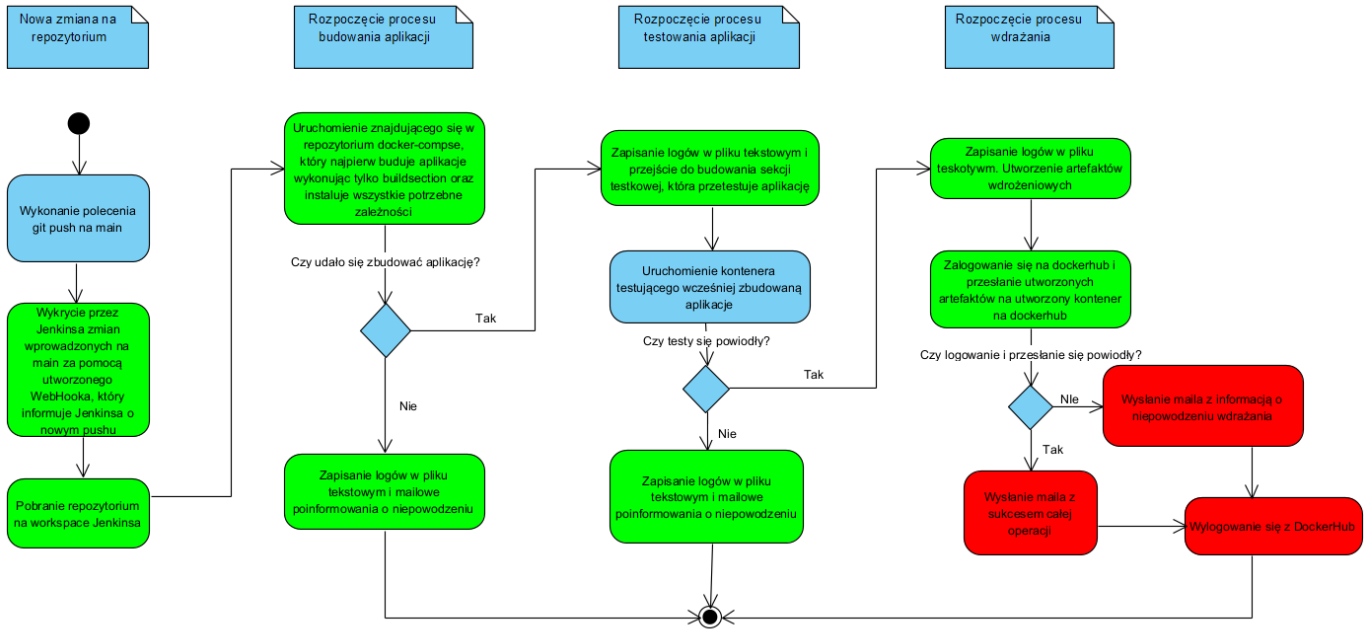
   Poniższy dokument przedstawia proces wdrażania komunikatora Deltachat (deltachat/deltachat-desktop: Email-based instant messaging for Desktop. (github.com)) za pomocą CI/CD pipeline i opisuje krok po kroku zaplanowane sekwencje kroków wdrożeniowych.

   **Technologie jakie zostaną użyte w omawianym wdrożeniu to:**

   a. Git, w tym GitHub i WebHook
   b. Docker, w tym Docker Compose
   c. Docker registry – DockerHub
   d. Jenkins, w tym Jenkinsfile
   e. Ngrok
   f. Serwer SMTP Google

   Do przygotowania planu wdrożenia zostało użyte oprogramowanie Visual Paradigm , w którym zostały przygotowane opisujące wdrożenie diagram aktywności i diagram wdrożenia

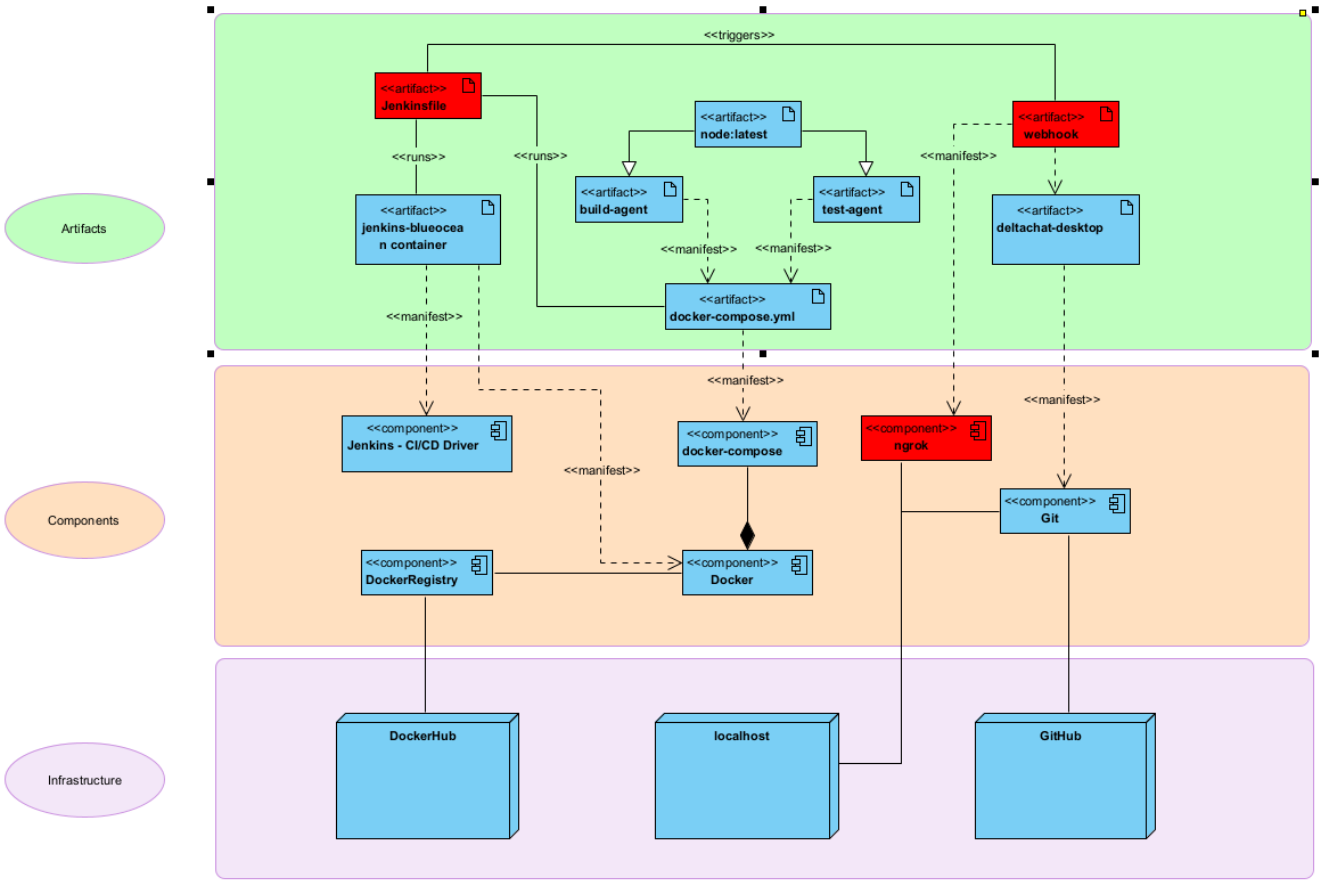
2. **Diagram aktywności ukazujący wszystkie czynności i główne decyzje podejmowane w zależności od występujących zdarzeń**




Kolor zielony oznacza zmienione aktywności a kolor czerwony nowo dodane aktywności

| Nazwa kroku | Technologia | Plik konfiguracyjny | Nr lini | Czy zgodnie z planem |
|---|---|---|---|---|
| Wykonanie polecenia git push na main | Git | - | - | Tak |
| Wykrycie przez Jenkinsa wprowadzonych zmian na main za pomocą WebHooka | Git - WebHook | - | - | Nie – Dodano obsługę WebHooka w celu informowania o nowych zmianach na repozytorium |
| Pobranie repozytorium na workspace Jenkinsa | Jenkins | Jenkinsfile | - | Tak + (zmiana komunikatu na diagramie) |
| Uruchomienie docker-compose do budowania aplikacji | Docker-compose Jenkins | Docker-compose.yml Jenkinsfile | 3… 5… | Nie – Uruchomienie tylko części docker-compose odpowiedzialnej tylko za budowanie aplikacji |
| Zapisanie logów i mailowa informacja o niepowodzeniu | Jenkins | Jenkinsfile | 9,15,21, 77-81 | Nie – Dodanie zapisywania logów do pliku tekstowego i dodanie informowania administratora o błędzie działania pipeline |
| Zapisanie logów i zbudowanie sekcji testującej z docker-compose | Docker-compose Jenkins | Docker-compose.yml Jenkinsfile | 8… 26… | Nie – dodanie zapisywania rezultatu budowania do logów i uruchomienie tylko sekcji dotyczącej testowania |
| Uruchomienie kontenera testującego wcześniej zbudowaną aplikacje | Docker-compose Jenkins | Docker-compose.yml Jenkinsfile | 8… 26… | Tak |
| Zapisanie logów i mailowa informacja o niepowodzeniu | Jenkins | Jenkinsfile | 31,37,43, 77-81 | Nie – Dodanie zapisywania logów do pliku tekstowego i dodanie informowania administratora o błędzie działania pipeline |
| Zapisanie logów i uruchomienie procesu tworzącego artefakty | Docker-compose Jenkins | Docker-compose.yml Jenkinsfile | 3… 53-55 | Nie – dodanie opcji zapisywania logów testowania w pliku |
| Zalogowanie się na dockerhub i przesłanie tam utworzonych artefaktów | Jenkins | Jenkinsfile | 56-58 | Nie – Dodanie opcji dodawania artefaktów na dockerhub |
| Wysłanie maila z informacją o niepowodzeniu wdrażania | Jenkins | Jenkinsfile | 64 | Nie – dodanie opcji wysyłania maili z błędami działania pipeline |
| Wysłanie maila z sukcesem całej operacji | Jenkins | Jenkinsfile | 79-80 | Nie – dodanie mailowego podsumowania działania pipeline |
| Wylogowanie się z dockerhub | Jenkins | Jenkinsfile | 78 | Nie – dodanie wylogowywania z dockerhub dla bezpieczeństwa |

## 3. Diagram wdrożeniowy




| Nazwa artefaktu/komponentu | Technologia | Plik konfiguracyjny | Nr lini | Czy zgodnie z planem |
|---|---|---|---|---|
| ngrok | ngrok | - | - | Nie – dodanie serwisu ngrok w celu poprawnej obsługi webhook |
| Docker-compose.yml | Docker-compose | Docker-compose.yml | - | Tak – tworzy i uruchamia obraz budujący i testujący |
| Jenkins-bluocean | Jenkins Docker | Dockerfile_jenkins | - | Tak – kontener Jenkinsa |
| Build-agent | Docker | Dockerfile_deltachat_build | - | Tak – obraz agenta budującego aplikacje |
| Test-agent | Docker | Dockerfile_deltachat_test | - | Tak – Obraz agenta testującego aplikacje |
| Deltachat-desktop | Git | Repozytorium | - | Tak – repozytorium z aplikacją |
| Jenkinsfile | Jenkins | Jenkinsfiles | - | Nie – dodanie artefaktu Jenkinsfile |
| node | node | - | - | Tak – zależność potrzebna do działania aplikacji |
| webhook | git | - | - | Nie – dodanie webhooka w celu automatycznego wywoływania pipeline |

4. **Omówienie czynności zaplanowanych do wykonania w ramach CI/CD pipeline**
    a. Rozpoczęcie wykonania pipeline nastąpi po wrzuceniu poleceniem git push nowym zmian na gałąź main. Kiedy zmiany zostaną wrzucone, przesłany zostanie web hook z githuba. Zostanie on wykryty przez Jenkins i automatycznie rozpocznie się proces budowania. Odpowiednie czynności będą wykonywane krok po k na podstawie przygotowanego Jenkinsfile.
    b. Po uruchomieniu procesu najpierw nastąpi sklonowanie repozytorium.
    c. Kiedy zakończy się klonowanie, rozpocznie się procedu budowania na podstawie istniejącego w repozytorium docker-compose.
    d. Pierwszy jest stage Build tak więc z pliku docker-compose zostanie wykonana tylko sekcja odpowiadająca za budowanie aplikacji.
    e. Jeśli budowanie aplikacji nie powiedzie się to nastąpi zakończenie wykonywania pipeline i zostanie wysłany email z logami z informacją o niepowodzeniu budownia
    f. Jeśli budowanie się powiedzie nastąpi przejście do kolejnego stage'a jakim jest testowania. W tym wypadku z docker-compose costanie zbudowania i utuchomiona sekcja odpowiedzialna za testowanie zbudowanej wcześniej aplikacji.
    g. Jeśli testowanie się nie powiedzie to tak jak wyżej wysyłana jest informacja mailem.
    h. Jeśli testowanie aplikacji powiodło się to możliwe jest przejście do ostatniego stage'a jakim jest utworzenie artefaktów wdrożeniowych aplikacji. Po ich utworzeniu artefakty zapisywane są w kontenerze, który jest przesyłany na dockerhub.
    i. Na końcu wysyłany jest mail z informacją czy wszystko się powiodło.