

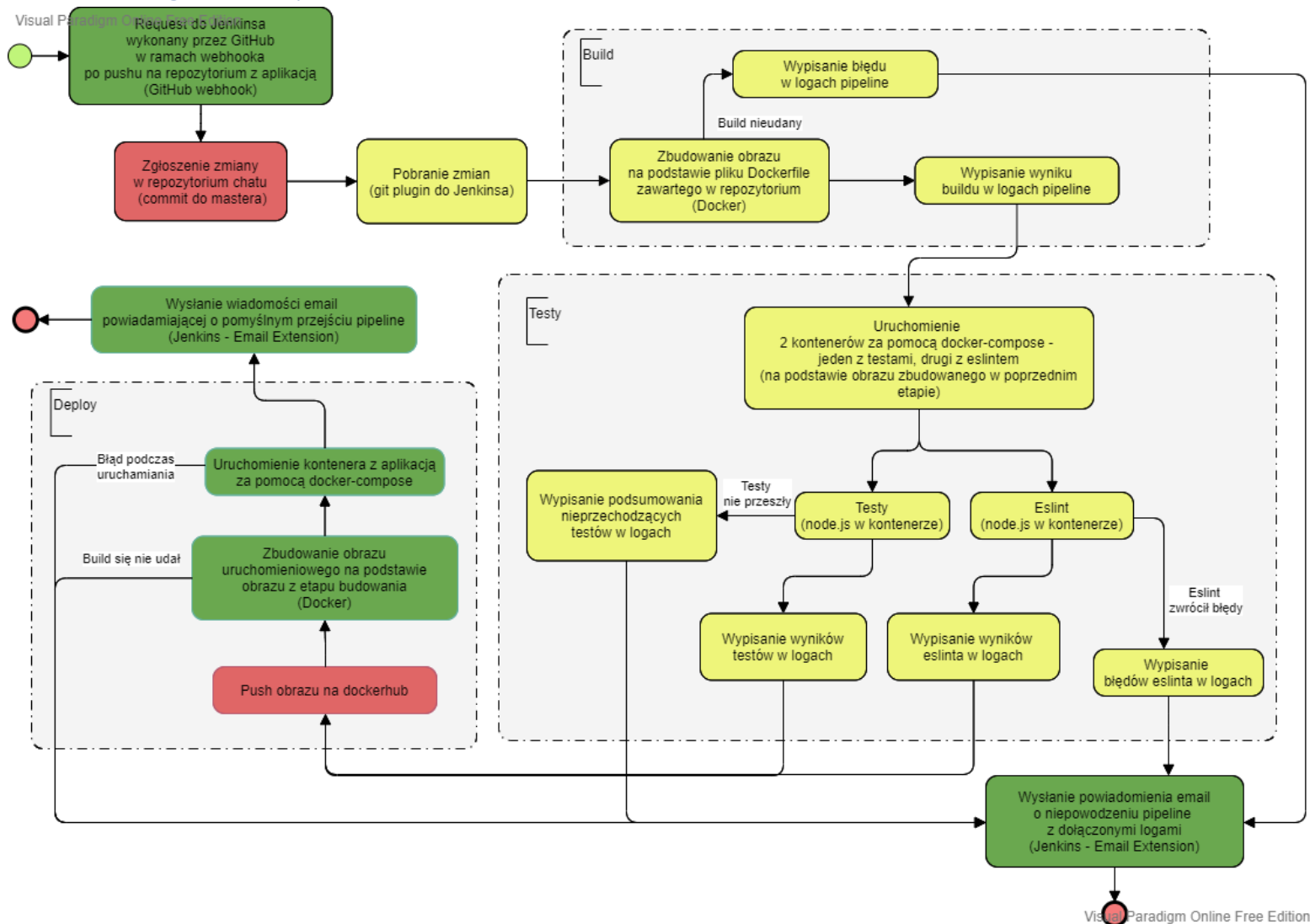
Omówienie implementacji planu wdrożeniowego aplikacji feathers-chat

Przygotował Maciej Adamus, 27.05.2021r

Wykorzystany stack technologiczny

- Docker
- Docker-compose
- Git
- Jenkins
- Ngrok

Diagram aktywności



Legenda:

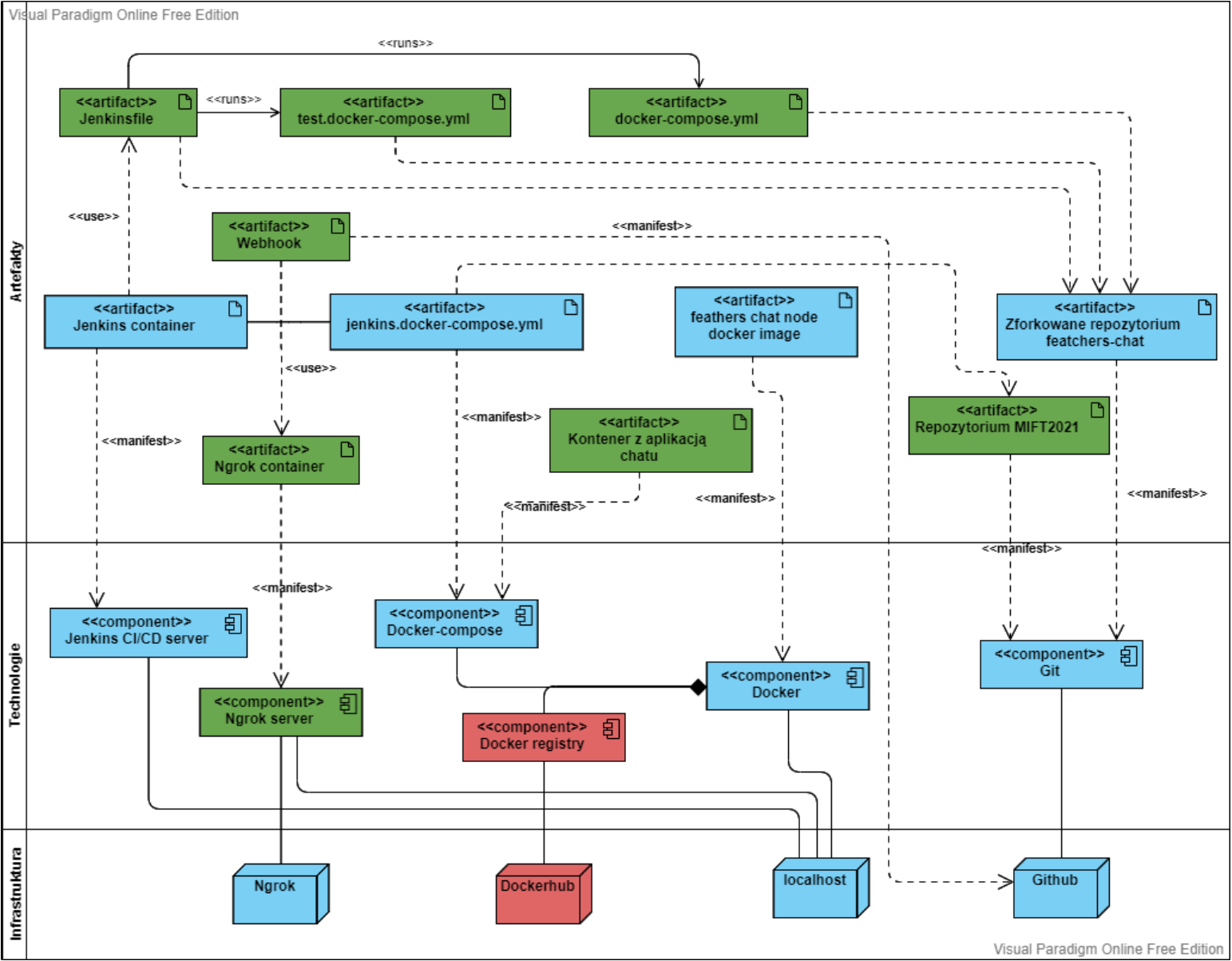
Czerwony – elementy pierwotnego planu które nie znalazły się w zaimplementowanej wersji

Zielony – kroki zaimplementowane, które nie były uwzględnione w pierwotnym planie pipeline

Nazwa kroku	Technologia	Linku do pliku konfigurującego (repo)	Czy zgodnie z planem + komentarz
Pipeline trigger by github webhook	Github, ngrok	ngrok setup Webhook skonfigurowany został za pomocą interfejsu githuba	Początkowy plan przewidywał wykrywanie zmian za pomocą mechanizmu SCM polling
Pobranie zmian z repozytorium	Git, Jenkins	Jenkinsfile	Zgodnie z planem
Zbudowanie obrazu na podstawie pliku build.Dockerfile	Docker, Jenkins	Jenkinsfile , Dockerfile	Zgodnie z planem
Wypisanie przebiegu buildu w logach pipeline	Jenkins	Automatycznie	Zgodnie z planem
W przypadku błędu przerwanie działania pipeline i wysłanie powiadomienia email z załączonymi logami	Jenkins	Jenkinsfile	Pierwotny plan nie przewidywał powiadomień email
Uruchomienie 2 kontenerów z testami i eslintem za pomocą docker-compose	Jenkins, docker-compose	Jenkinsfile , docker-compose	Zgodnie z planem
Wypisanie podsumowania nieprzechodzących testów w logach w przypadku niepowodzenia	Jenkins	Automatycznie	Zgodnie z planem
Wypisanie błędów zwróconych przez eslint	Jenkins	Automatycznie	Zgodnie z planem
W przypadku błędu przerwanie działania pipeline i wysłanie powiadomienia email z załączonymi logami	Jenkins	Jenkinsfile	Pierwotny plan nie przewidywał powiadomień email
Push obrazu na dockerhub	Jenkins, docker	-	Finalny pipeline nie przewiduje publikowania obrazów
Zbudowanie obrazu uruchomieniowego na podstawie obrazu buildowego	Jenkins, dockers	Jenkinsfile , Dockerfile	Pierwotny plan jako deploy przewidywał opublikowanie obrazu ze zbudowaną aplikacją na dockerhub
W przypadku błędu przerwanie działania pipeline i wysłanie powiadomienia email z załączonymi logami	Jenkins	Jenkinsfile	Pierwotny plan nie przewidywał powiadomień email
Deploy na podstawie zbudowanego obrazu z aplikacją – docker-compose	Jenkins, docker-compose	Jenkinsfile , docker-compose.yml	Do deployu użyty został docker-compose w celu ułatwienia ewentualnego przyszłego rozszerzania infrastruktury która zostaje uruchomiona w ramach deployu

Wysłanie email'a z powiadomieniem o powodzeniu wykonania pipeline	Jenkins	Jenkinsfile	Pierwotny plan nie przewidywał powiadomień email
---	---------	-----------------------------	--

Deployment diagram



Nazwa artefaktu	Technologia	Link do pliku w repo	Czy zgodnie z planem + komentarz
Jenkins container	Docker, Jenkins	Jenkins dockerfle , Jenkins docker-compose	zgodnie z planem
Jenkinsfile	Jenkins	Jenkinsfile	Pierwotny diagram nie przewidywał użycia Jenkinsfile bo w momencie jego tworzenia nie wiedziałem że można to tak rozwiązać
Webhook	github	-	Pierwotny plan nie przewidywał użycia webhooka

Ngrok container	Ngrok	Ngrok	Umożliwia uruchamianie buildów za pomocą webhooka przez utworzenie tunelu pozwalającego na dostęp do localhosta „z zewnątrz”
test.docker-compose.yml	docker-compose	docker-compose	Pierwotny diagram nie uwzględniał tego artefaktu
Jenkins.docker-compose.yml	docker-compose	docker-compose	Zgodnie z planem
Kontener z aplikacją chatu	Docker	Dockerfile	Pierwotny plan nie uwzględniał tego artefaktu
feathers-chat app docker image	Docker	Dockerfile	Zgodnie z planem
Repozytorium MIFT2021	Git	-	Pierwotny plan nie uwzględniał tego artefaktu, ponieważ jako swoje single source of truth przyjąłem zforkowane repozytorium z aplikacją chatu
docker-compose.yml	docker-compose	docker-compose.yml	Pierwotny plan nie uwzględniał tego artefaktu
Zforkowane repozytorium feathers-chat	Git	-	Zgodnie z planem