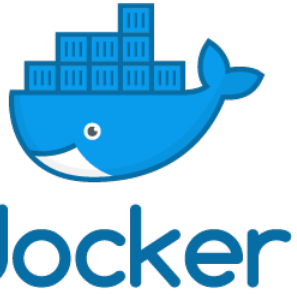
Универзитет „Св. Кирил и Методиј“ - Скопје
Факултет за информатички науки и компјутерско инженерство

Continuous Integration / Continuous Deployment на веб апликација со помош на Jenkins.



Jenkins

Изработил: Небојша Тодоровиќ 209015

Веб апликација

- Приказ на информации за возачи и тимови во Формула 1
- API (Java Spring Boot) кое зема податоци од DBPedia, со помош на Sparql queries, нуди повеќејазичност, пребарување по некој текст, и линкови кон надворешни страни. Достапно и за нов front-end.
- Front-end (AngularJs) приказ на информациите и комуникација со API.

CI/CD

- Continuous integration. Интеграција на работата од повеќе девелопери дури и неколку пати на ден.
- Continuos delivery гранката која е наменета за продукција е секогаш спремна за достава.
- Continuous deployments, доставата се врши автоматски при одредени услови, не мора да е на продукција.

Jenkins

- Слуша повици од Github webhooks, наместени на гит.
- Извршува наредби зададени во Jenkinsfile.
- Наредбите се извршување на тестови, компајлирање на апликациите, известување на гит за резултати од истите (pending, success, failed).
- Сите операции се прават во посебни докер контејнери, и доколку успешно се извршат тестовите и гранката е develop се повикува скрипта која прави автоматски deploy на девелопмент околината.

Ansible

- Искористен за конфигурирање на серверот и достава на апликацијата.
- Две скрипти (playbooks) една за достава на продукција, една за development околина.
- Скриптата за достава на продукција може да се искористи за автоматска достава со Jenkins.
- Тоа што истата скрипта се грижи и за конфигурација на серверот овозможува лесна промена на физичката машина на која е доставена апликацијата (се конфигурира само нов/друг хост).

NGINX

- Load balancer, proxy (development, production, Jenkins).
- Овозможува лесно скалирање поради можноста за промена на конфигурација во runtime (не е имплементирано во овој проект, но ќе биде идно подобрување)

Docker

- Искористен за креирање контејнери, во кои има стартувано сервиси (production/development front-end & back-end, jenkins, nginx, контејнери кои ги креира и брише jenkins).
- Овозможува опслужување на повеќе сервиси на иста физичка машина, кои работат во своја околина.
- Контејнерите овозможуваат минимален down-time, бидејќи користат локална меморија од хостот и потребен е само рестарт за да биде достапната верзија. Има и техника за достава без down-time.
- Искористен е docker-compose за полесно менаџирање на контејнерите.

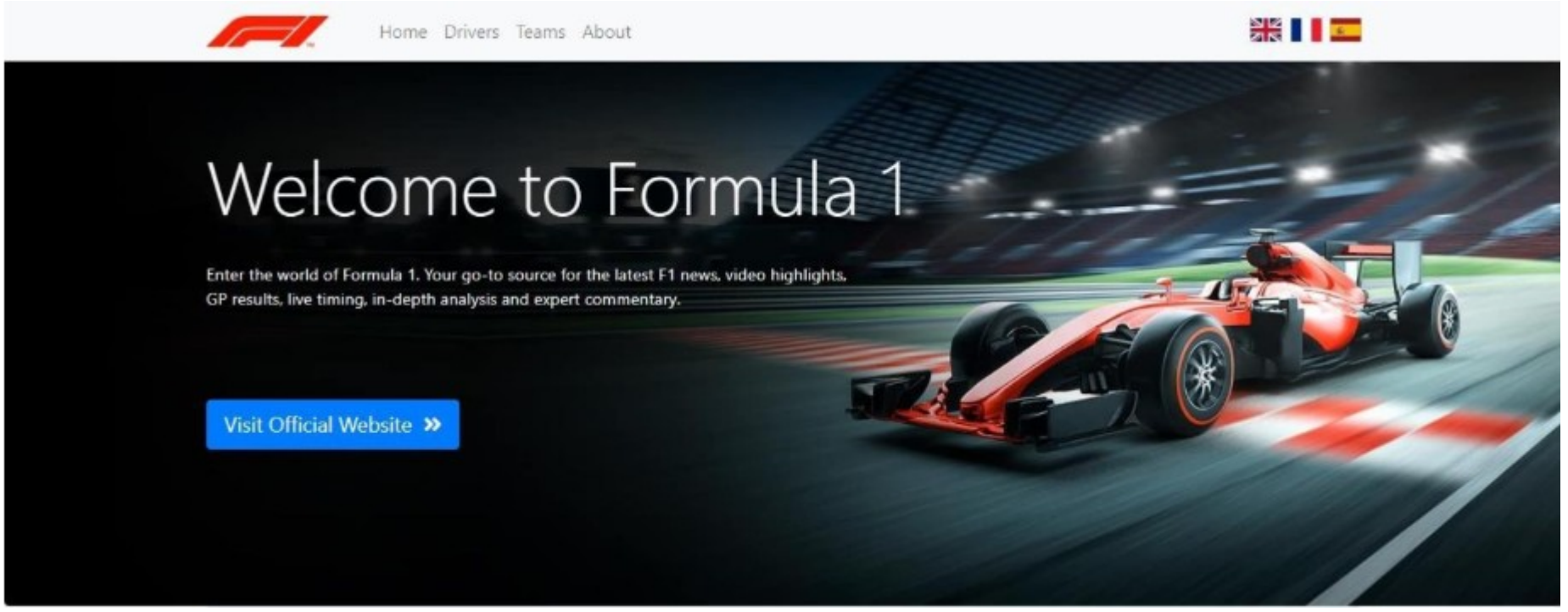
Споредба на контејнери во однос на виртуелни машини

- Виртуелни машини се побезбедни
- Минимум 2gb RAM (препорачливо е, само за машината + за апликацијата. Минимум 8gb меморија (препорачливо е 20gb) за виртуелна машина. Целиот Docker систем бара минимум 512 MB RAM за да функционира, меморија зависно од контејнерите. Возможна е динамичка алокација и ограничувања. Овој проект има 6-7 контејнери кои се активни на машина со само 8GB RAM и 30GB меморија, со процесор со 2 јадра.
- Полесно се врши верзионирање со контејнери, што олеснува враќање назад на претходна верзија при катастрофална грешка. Со виртуелни машини овој процес е покомплициран, поради нови верзии на библиотеките, кои не се изолирани како што тоа е случај кај контејнерите.
- Полесно се покренува инстанца од контејнер отколку виртуелна машина и поради помалата комплексност истиот процес се извршува побрзо.

Понатамошна разработка на проектот

- Consul за информации за контејнерите
- Динамичка конфигурација за Nginx
- Blue-green deployments со 0 downtime
- Опслужување во Kubernetes/Swarm кластер, полесно скалирање
- Health checks & recovery

Demo


Home
Drivers
Teams
About
Welcome to Formula 1
Enter the world of Formula 1. Your go-to source for the latest F1 news, video highlights,
GP results, live timing, in-depth analysis and expert commentary.
Visit Official Website »

Прашања?