Dokumentacija - Sistemska administracija

Skupina: OnlyFriends/Kolesarstvo v Mariboru

1. **Uvod**

Pod okriljem Projektnega praktimuma v sklopu 4 predmetov (Sistemska administracija, Principi programskih jezikov, Prevajanje programskih jezikov in Spletno programiranje) smo morali ustvariti skupinski projekt. Naša skupina se je odločila za aplikacijo namenjeno kolesarjem.

1. **Načrtovane naloge**
   1. **Kaj je bilo potrebno storiti**

Za prvo projektno nalogo pri predmetu Sistemska administracija smo morali vzpostaviti našo aplikacijo narejeno pri Spletnem programiranju na platformi Docker in na storitvi Heroku. Pri drugi projektni nalogi je pa bilo potrebno spoznati ukaze za pregled in upravljanje naše aplikacije na storitvi Heroku.

* 1. **Katere projektne naloge smo izbrali**

Za reševanje teh dveh projektnih nalog smo namestili Docker in vsi člani ustvarili račun na Heroku, Za našo podatkovno bazo smo uporabili MongoDB Atlas, zastonj storitev v oblaku

1. **Implementacija**

V naši aplikaciji smo ustvarili datoteko z imenom »Dockerfile«, kjer je bilo potrebno zapisati določene stvari za ustvarjanje slike naše aplikacije na platformi Docker. Ta slika se je ustvarila z ukazom »docker build . -t >ime\_slike<« in sam zaboj se je kreiral z ukazom »docker run –p PORT:PORT >ime\_slike<« in še eno datoteko ».dockerignore« v katero smo zapisali datoteke, ki jih naj naš »Dockerfile« ignorira pri ustvarjanju slike. S tem smo skrajšali čas ustvarjanja slike in pohitrili našo aplikacijo.

Na Heroku je naš vodja moral ustvariti novo aplikacijo, na katero smo s pomočjo »Container Registry« namestili našo aplikacijo. Pri tem nam je v pomoč prišel Docker, saj lahko z »Dockerfile« namestiš aplikacijo tudi na storitvi Heroku. Prvo smo morali v terminalu naše aplikacije uporabiti ukaz »heroku login«, da smo se prijavili v naš Heroku račun, kjer smo lahko naložili našo aplikacijo.

Sledil je ukaz »heroku container:login«, s tem smo omogočili, da lahko naložimo Docker slike na Heroku. Z ukazom »heroku container:push web –>ime\_ aplikacije<« smo sliko iz Dockerja zgenerirali in potisnili v »Container Registry«, nato pa z ukazom »heroku container:release web ->ime\_aplikacije<« spustili to sliko v našo aplikacijo ustvarjeno na storitvi Heroku.

Za drugo projektno nalogo smo uporabili ukaz »heroku logs –tail«, s katerim smo pridobili izpise naše aplikacije. Z ukazom »heroku maintenance:on« in »heroku maintenance:off« smo vklopili in izklopili vzdrževalni način aplikacije. Ukaz »heroku config:set 'ime'='vrednost'« omogoča shranjevanje okoljskih spremenljivk v okolju Heroku. Ime aplikacije se lahko spremeni s pomočjo ukaza »heroku apps:rename >novo\_ime< --app >staro\_ime<« Ukaz »heroku domains« smo uporabili za izpis vseh domen naše aplikacije. Z ukazom »heroku domains:info >domena<« lahko pridobimo podrobnosti o tej domeni. Za ustvarjanje nove domene lahko uporabimo ukaz »heroku domains:add >nova\_domena<«. Za zadnji del naloge je bilo potrebno preveriti kakšen tip kreditnih Dynos uporablja naša aplikacija in kako bi lahko te skalirali. Ugotovili smo, da gre za zastonj kreditne točke Dynos in da bi lahko našo aplikacijo skalirali na štiri različne načine.

1. **Zaključek**

Skozi ta projekt smo se naučili kako deluje Docker in Heroku. Kaj sploh so slike in zaboji na Docker-ju. Kako deluje sama platforma kot storitev Heroku in za kaj je uporabno. Prav tako smo se naučili uporabljati storitev Jira za vodenje projektov in razdelitev nalog.

1. **Viri:**

Namestitev Docker-ja:

<https://docs.docker.com/desktop/windows/install/>

Heroku storitev:

<https://www.heroku.com/home>

MongoDB Atlas:

<https://www.mongodb.com/atlas/database>

Pomoč za ustvarjanje Dockerfile:

<https://docs.docker.com/engine/reference/builder/>

Heroku Container Registry za namestitev aplikacije:

<https://devcenter.heroku.com/articles/container-registry-and-runtime>

Heroku izpisi:

<https://devcenter.heroku.com/articles/logging>

Vzdrževalni način Heroku:

<https://devcenter.heroku.com/articles/maintenance-mode>

Heroku okoljske spremenljivke:

<https://devcenter.heroku.com/articles/config-vars>

Heroku sprememba imena aplikacije:

<https://devcenter.heroku.com/articles/renaming-apps>

Heroku domene:

<https://devcenter.heroku.com/articles/custom-domains>

Heroku kreditne točke Dynos in skaliranje:

<https://devcenter.heroku.com/articles/dynos>