PEREB STYLE  
  
Lucrarea de laborator Nr.1

Obiectiv: Crearea mediului pentru dezvoltarea aplicații cloud native

Descriere:

1. Se va descarcă si instala mediul Rancher Desktop care va permite crearea containerilor si

componentelor Kubernetes.

Link: https://docs.rancherdesktop.io/getting-started/installation/

2. Se va familiariza cu funcționalitățile disponibile in Rancher Desktop utilizând tutorial si ghiduri

disponibile la : https://docs.rancherdesktop.io/

3. Se va identifica sau elabora o aplicație care va fi utilizata pe parcursul lucrărilor de laborator.

aici deja instalezi Rancher Desktop  
  
Lucrarea de laborator Nr.2

Obiectiv: Familiarizarea cu procesul de creare a containerilor

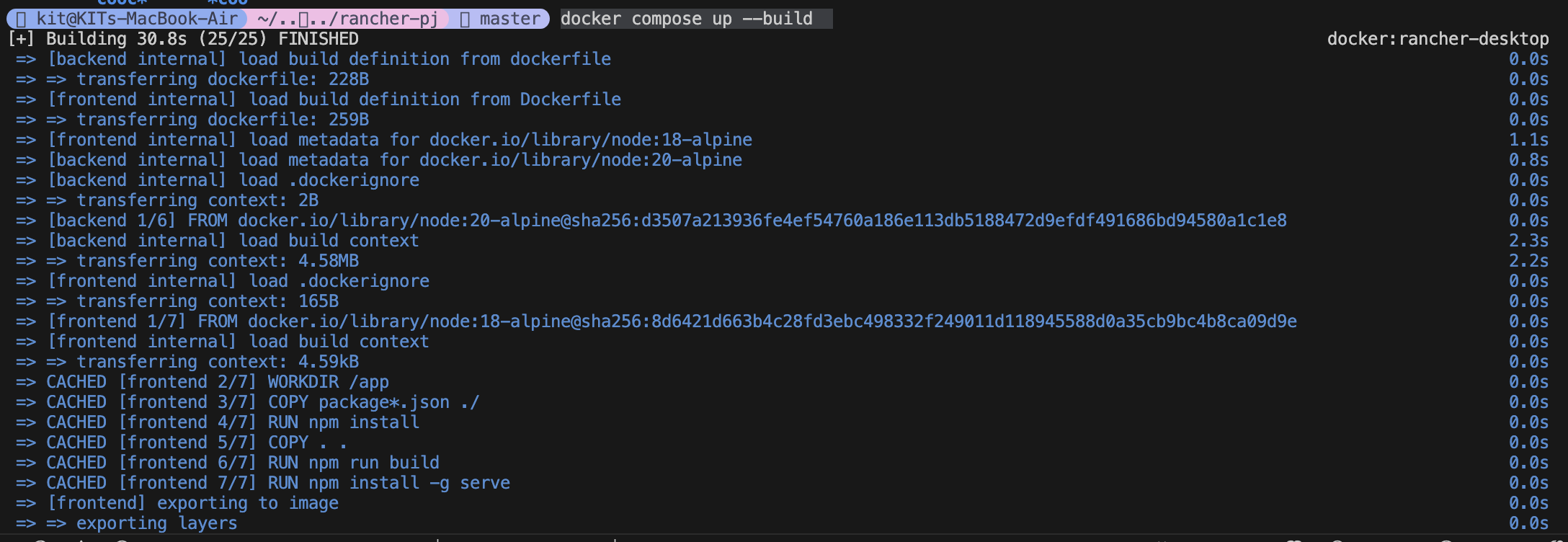
Descriere:

1. Utilizând Rancher Desktop , aplicația creata/identificata in lucrarea de laborator N.1 va fi

containerizata.

1. Containerul creat va fi lansat si va fi demonstrat ca aplicația poate fi accesata pentru a verifica

funcționalitatea.

-pe partea front (SneakersR) cream dockerfile pentru a define containereu .  
-pe partea de back (sneakers-back) cream dockerfile la fel   
  
apoi in root folder docker-compose – pentru a rula simultan containerele .   
  
rulam ambele containere local cu comanda : docker compose up –build -  
din directoriul root (rancher-pj) – asa se cheama folderu   
  
  
  
ca rezultat containerele sunt create ( 2 ) back si front -  
fontu accesibil la. <http://localhost:3000/>   
iar backendu la <http://localhost:5002/>   
  
Lucrarea de laborator Nr.3

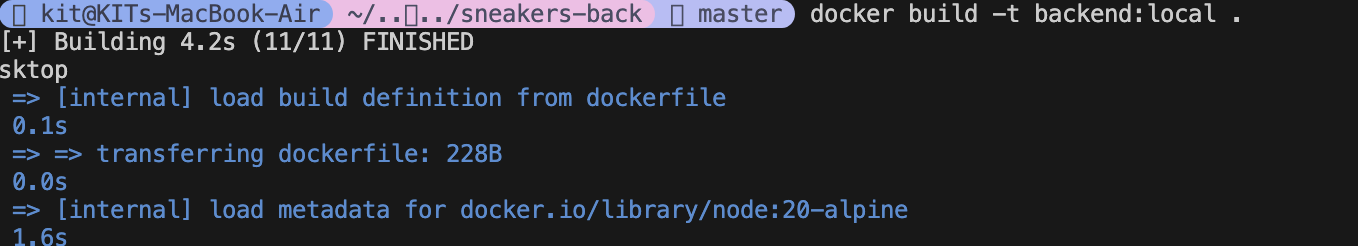
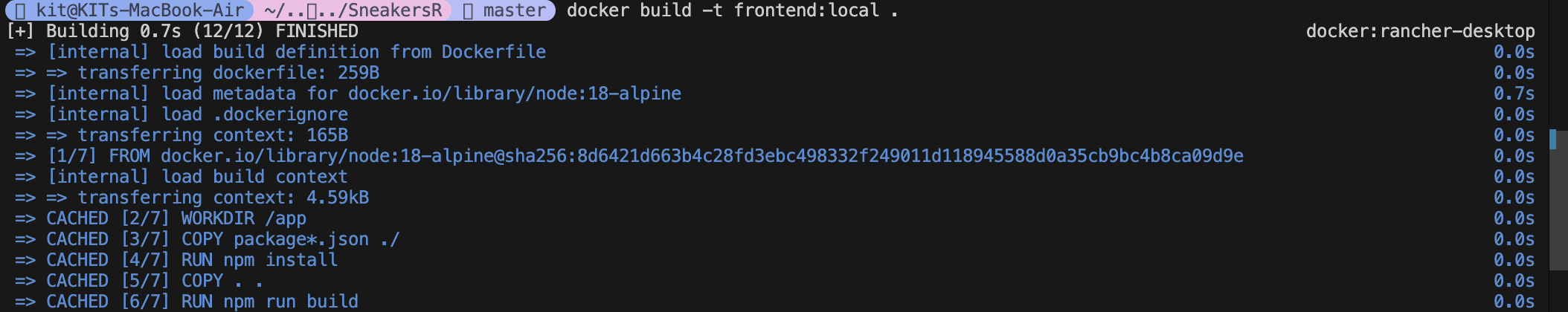
Obiectiv: Familiarizarea cu procesul de creare a obiectelor Kubernetes

Descriere:

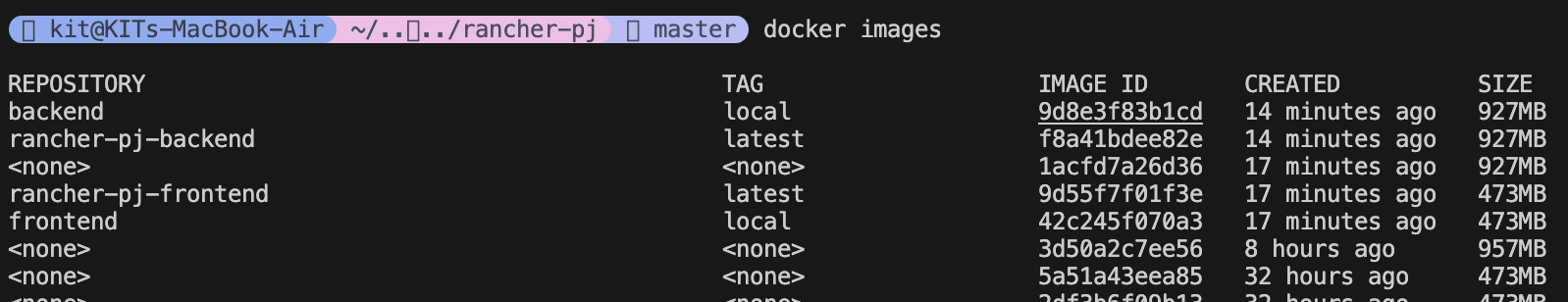
1. Utilizând Rancher Desktop , pentru aplicația creata/identificata in lucrarea de laborator N.1 vor

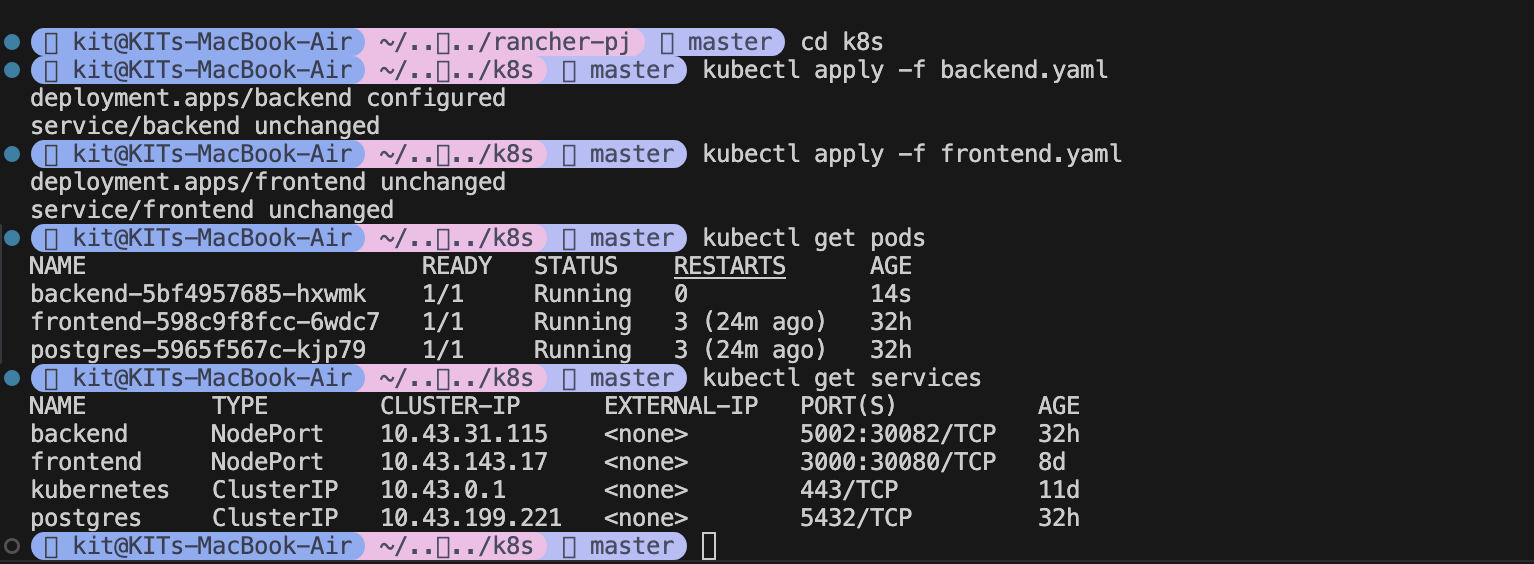
fi create obiectele Kubernetes necesare.

2. Aplicația va fi lansata pe platforma Kubernetes si accesata pentru a verifica funcționalitatea .  
  
  
CREAM imaginile pentru fiecare

Verificam imaginile create



Cream folderu k8s in root folder pentru Kubernetes   
cu fisierele de configurare pentru front si back :  
  
  
ne mutam cu cd in folderu k8s si aplicam modificarile   
  
apoi cu ajutorul :  
kubectl get pods, kubectl get service verificam daca tot merge si s au creat .(startate )   
  
 <http://localhost:30080> → Frontend

 <http://localhost:30082> → Backend  
  
  
Lucrarea de laborator Nr.4

Obiectiv: Familiarizarea cu procesul de creare a flow-urilor CI/CD

Descriere:

1. Vor fi studiate serviciile Google Cloud care sunt puse la dispoziție de Google pentru crearea flow

2. CI/CD

Va fi implementat flow de CI/CD pe platforma Google Cloud pentru a face un deployment a

aplicației create/identificate in lucrarea de laborator N.1, pe Cloud Run si/sau GKE.

Lucrarea de laborator Nr.4

Obiectiv: Familiarizarea cu procesul de creare a flow-urilor CI/CD

Descriere:

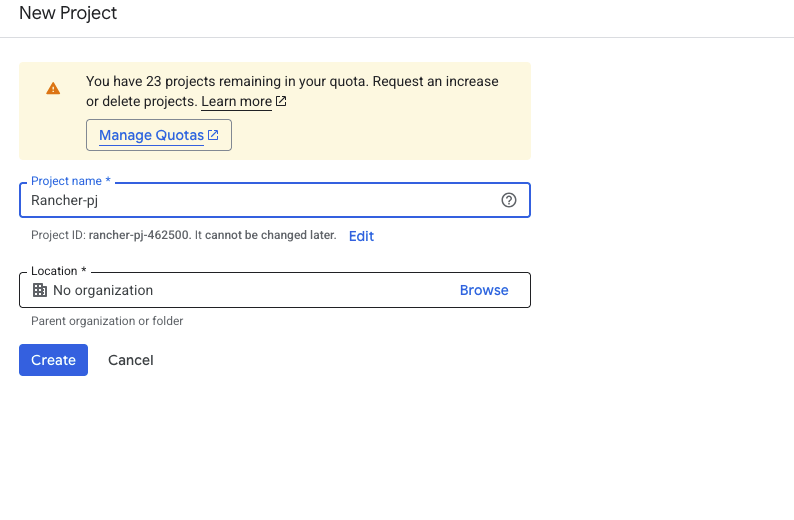
1. Vor fi studiate serviciile Google Cloud care sunt puse la dispoziție de Google pentru crearea flow

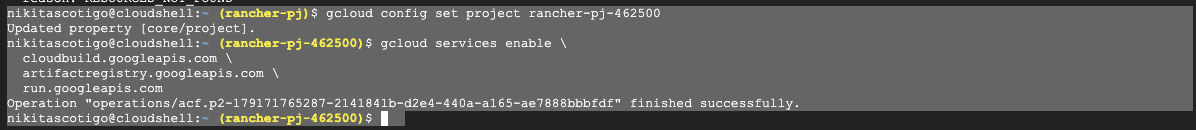
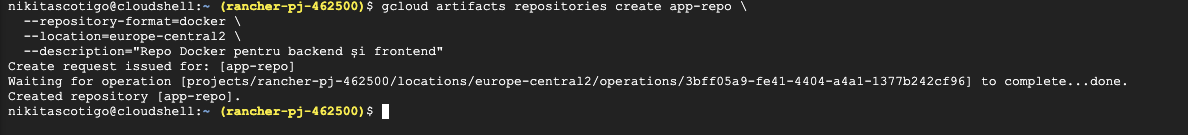
2. CI/CD

Va fi implementat flow de CI/CD pe platforma Google Cloud pentru a face un deployment a

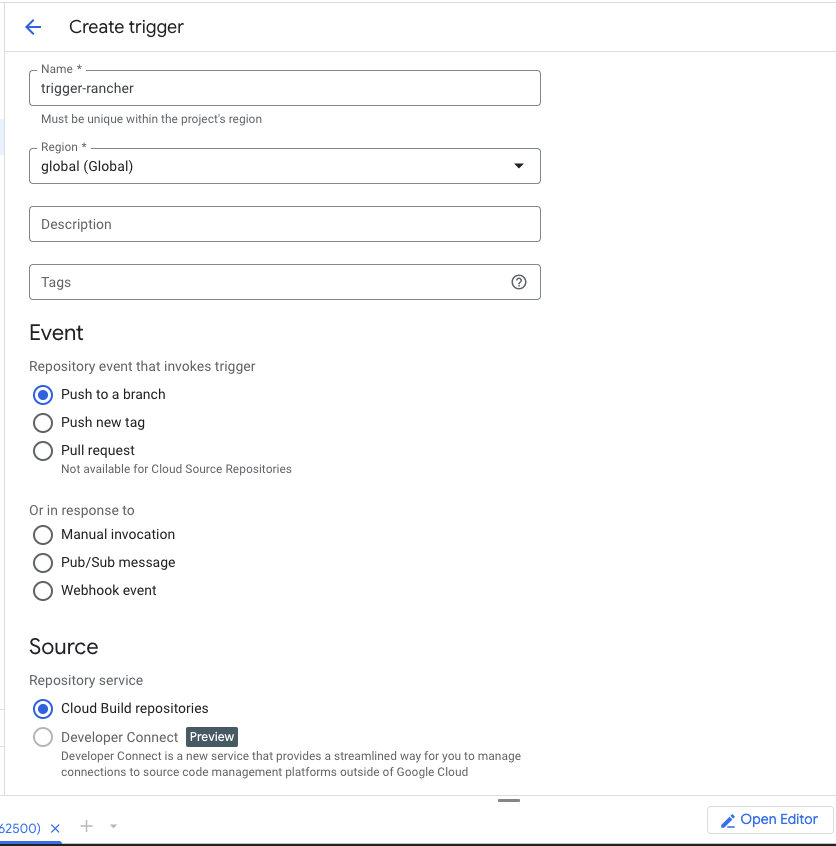
aplicației create/identificate in lucrarea de laborator N.1, pe Cloud Run si/sau GKE.

Cream un proiect nou in google cloud console :

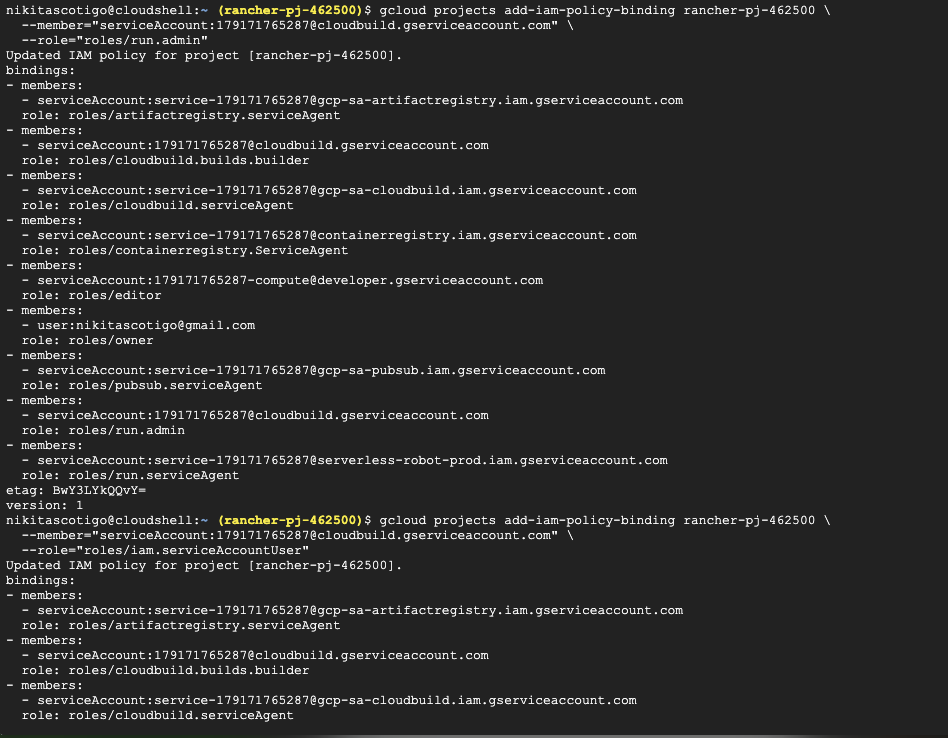


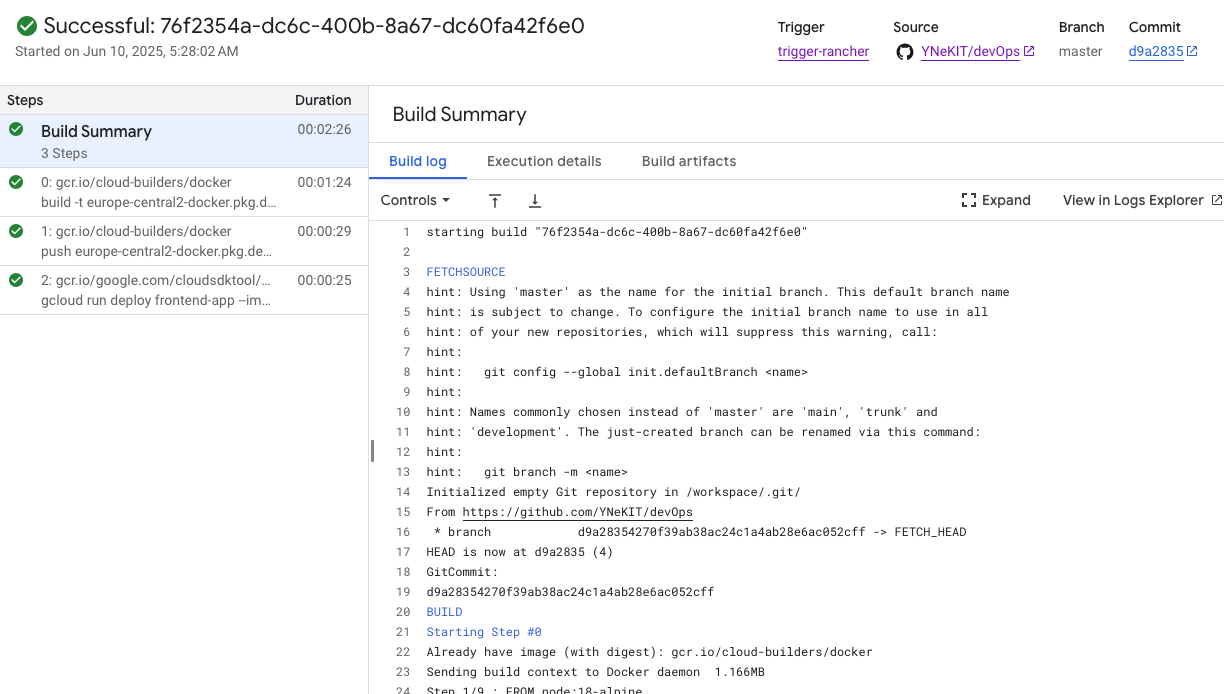
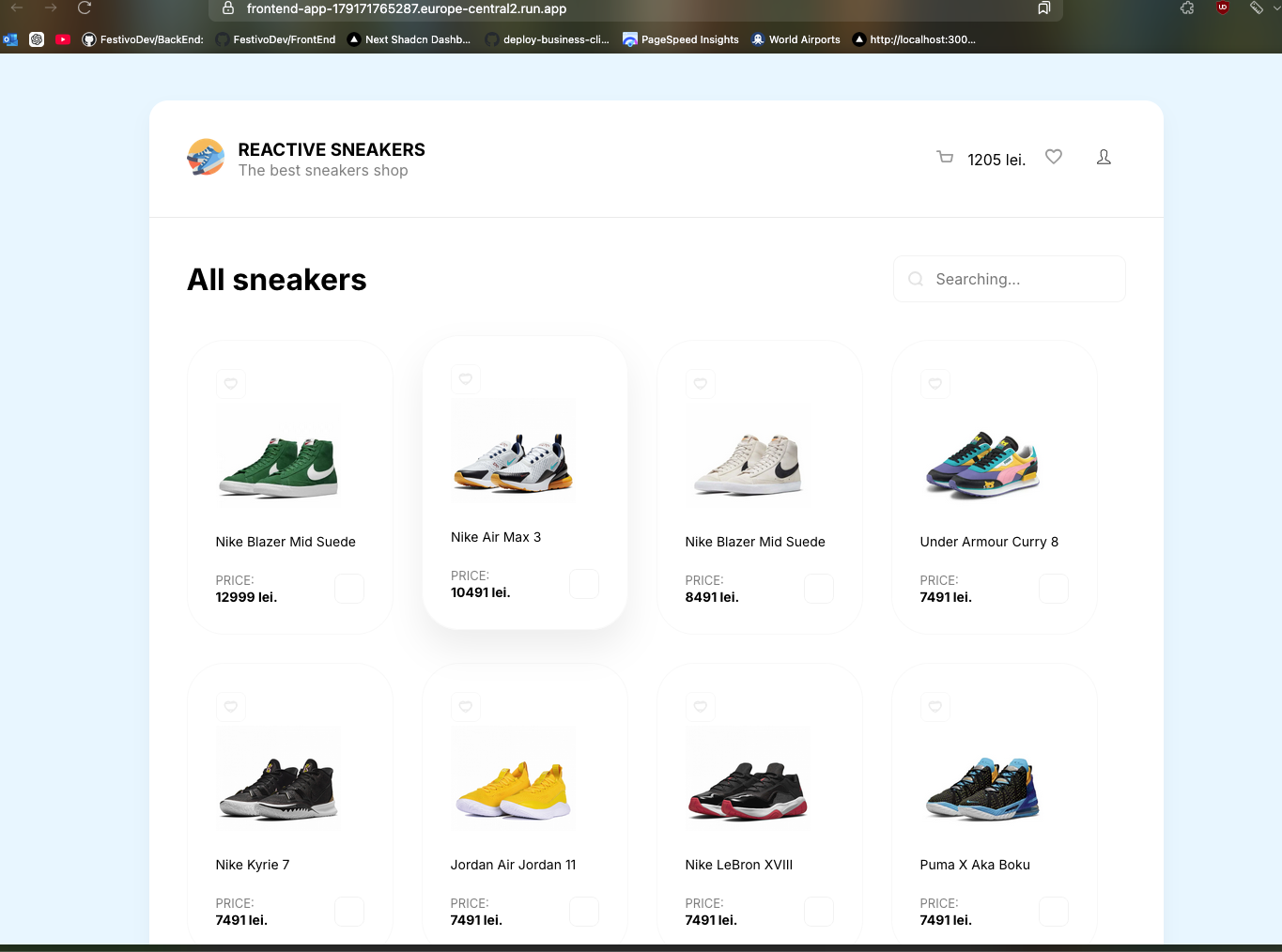
Setam tot zbs in cloud shell :  
  
  
  


Cream un trigger din build triggers.  
conectam repo de la github ( permitem accesul )



Acordam niste permisiuni ca google cloud se ahueste la build

  
  
  
pana la urma numa frontu merge ( da rasta e de ajuns ! ) rabotaet I hui s nim .  
deci cand faci vreo modificare in SNeakersR adaugi ceva text etc , faci git add .   
git commit -m “ceva coment “ dupa git push origin main si se incarca pe github de acolo este conenctat cu google cloud RUN si se face trigger la event. Si se incepe buildu .  
  


  
  
asta e build cu success. El poate dura vreo 3-4 min pana se creaza.  
  
dupa o sa ai un link de acces unde poti vizualiza pe web situ   
spre exemplu : https://frontend-app-179171765287.europe-central2.run.app/  
  
  
  
  
produsu final   
  
