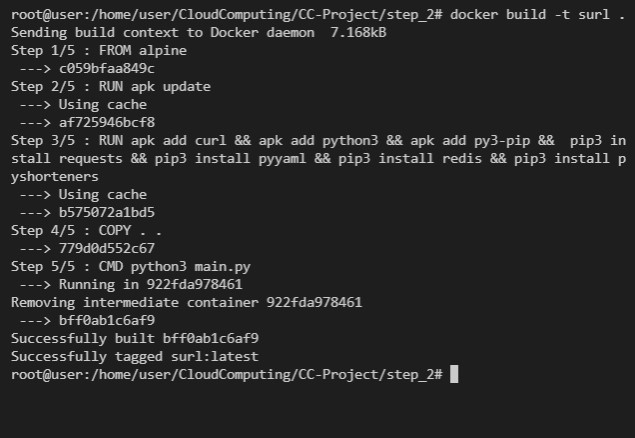
|  |  |
| --- | --- |
| **سید مهدی میرفندرسکی** | **محمدمهدی هجرتی** |
| **9723093** | **9723100** |
| **پروژه ی پایانی رایانش ابری** | **داکر و کوبرنتیز** |
| **استاد جوادی** | **بهمن 1400** |

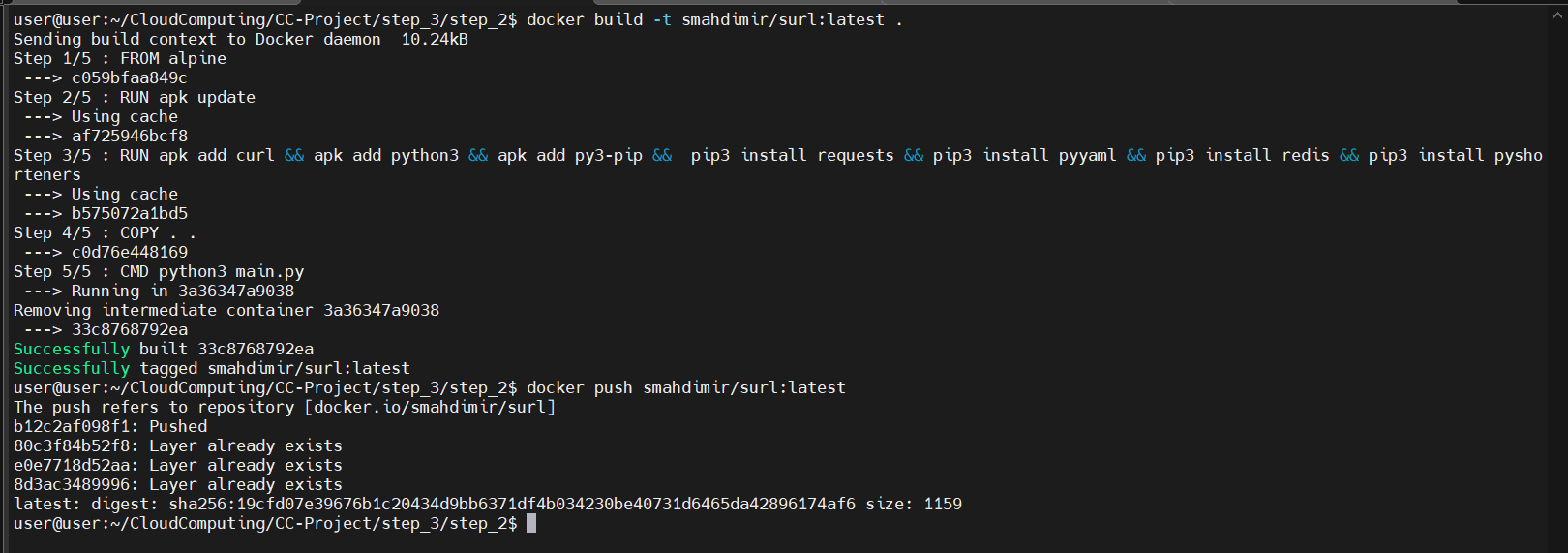
**به نام خدا**

**گام دوم:**

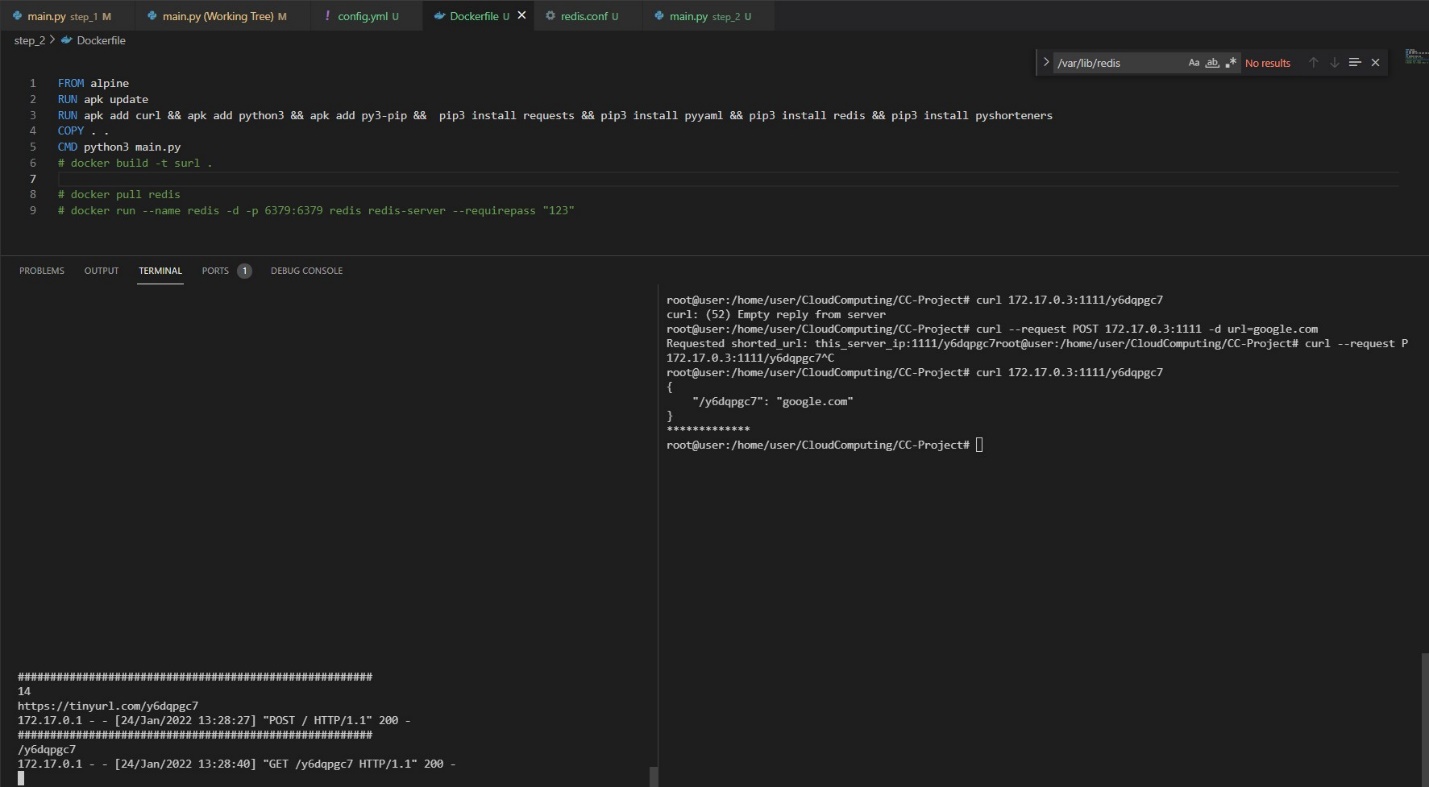
1. بیلد کردن ایمیج



1. ارسال ایمیج روی داکرهاب



1. تست روی سیستم شخصی



1. محتویات داکرفایل

FROM alpine

RUN apk update

RUN apk add curl && apk add python3 && apk add py3-pip &&  pip3 install requests && pip3 install pyyaml && pip3 install redis && pip3 install pyshorteners

COPY . .

CMD python3 main.py

**گام سوم:**

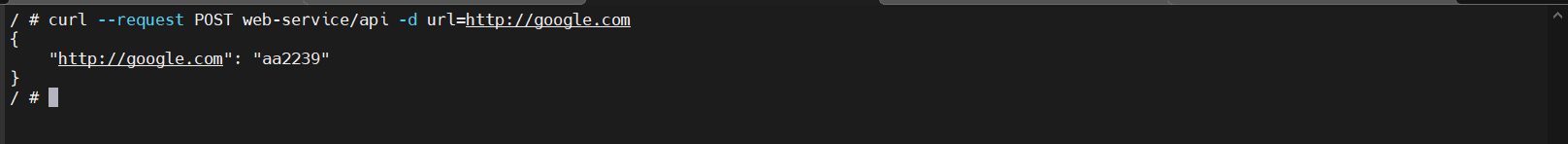
بررسی منابع:

Text

Description automatically generated

برای جلوگیری از ناسازگاری داده‌ها از یک پاد ردیس استفاده شده است. در صورت استفاده از چندین پاد مشکلات متعددی در درج و خواندن پیش خواهد آمد.

تست سیستم:



Background pattern

Description automatically generated

بعد از گذشت زمان انقضا:

Graphical user interface

Description automatically generated with medium confidence

درخواست نامعتبر:

Text

Description automatically generated

**موارد امتیازی:**

**ساخت کامپوننت HPA**

1. به طور کلی ۳ نوع پارامتر برای مقیاس کردن خودکار وجود دارد. که resource metric و pod metric و object metric می‌باشد. در resource metric میانگین میزان مصرف شده ی cpu و memory قابل بررسی می‌باشد به صورتی که اگر از مقدار درصد تعیین شده کمتر یا بیشتر مصرف شده باشد تعداد پاد ها به اندازه مورد نیاز تغییر می‌کند. در pod metric میزان استفاده از منابع در پادها مورد بررسی قرار می‌گیرد. در روش سوم معیار بر روی object های متفاوتی تعریف می‌شود.
2. ما از resource metric ها و به طور دقیق‌تر از مصرف cpu برای ایجاد HPA استفاده کردیم. با توجه به اینکه در اینجا صرفاً کارکرد درست برنامه مد نظرمان بود و میخواستیم در صورتی که از بیش از ۵۰ درصد cpu پاد ها استفاده شد، به طور خودکار منابع جایگزین آماده شوند.
3. با دستور زیر hpa را می سازیم:

kubectl autoscale deployment web-deployment.yaml --cpu-percent=60 --min=2 --max=8

با دستور زیر نیز می‌توان جزئیات آن را مشاهده کرد:

kubectl get hpa

**اجرا با statefulset و جایگزینی با deployment**

1. به طور کلی برنامه‌های stateful برنامه‌هایی هستند که داده‌ها را نگه می‌دارند و مانند دیتابیس ها تغییرات آن‌ها را ترک می‌کنند. در مقابل stateless ها برنامه‌هایی هستند که مانند nginx و … داده‌ای در خود نگهداری نمی‌کنند و هر بار داده ی جدید گرفته و آن را پردازش می کنند. در این برنامه ها نیاز به پاد های با identity منحصر به فرد وجود دارد به طوری که هر پاد­ از پاد های دیگر قابل شناسایی باشد. به علاوه در statefulset حتماً به persistent storage داریم در این حالت در صورت نابود شدن یک پاد ولیوم اختصاص داده شده باقی می ماند. به همین دلیل در اینجا نیز برای استفاده از ردیس از statefulset استفاده می‌کنیم.

**پیاده سازی docker-compose**

فایل docker-compose برای خودکار سازی فرآیند ایجاد دو سرویس redis و web ساخته شد که در آن آدرس قرارگیری Dockerfile یا ایمیج مورد نیاز و تمام نیازمندی های به کار رفته شامل سکرت ها و ولیوم های اختصاص داده شده و متغیر های محیطی و پورت های مپ شده و سایر اطلاعات آورده شده است.