



دانشگاه صنعتی امیرکبیر  
( پلی تکنیک تهران )

# مبانی رایانش ابری

تمرین دوم (فاز ۰ و ۱)

داکر

طراحان تمرین:

علی بازشوشتری، ابوالفضل حسینی

استاد درس:

دکتر جوادی

مهلت نهایی ارسال پاسخ:

۱۷ فروردین ساعت ۲۳:۵۹

## مقدمه

هدف از این تمرین، کار با داکر<sup>1</sup> است. بنابراین در این تمرین یک پروژه بسیار ساده را با استفاده از داکر، کانترینایز<sup>2</sup> می‌کنید. برای پیش نیاز، لازم است داکر را بر روی سیستم خود نصب کرده باشید. برای راهنما می‌توانید از لینک‌های زیر استفاده کنید. دقت داشته باشید که برای اتصال به dockerhub، نیاز به تنظیم شکن یا استفاده از فیلترشکن دارید.

[Get Docker | Docker Docs](#)

---

<sup>1</sup> Docker

<sup>2</sup> Containerize

## فاز ۰

### بخش ۱

در ابتدا باید در [داکرهاب](#) يك اكانت بسازيد.

در این بخش شما بایستی فعالیت‌های زیر را به ترتیب انجام دهید:

1. در محتویات تمرین یک فایل با نام cc\_hw2\_hello.py در اختیار شما قرار گرفته است. این فایل حاوی یک

قطعه کد پایتون می‌باشد که در آن از پکیج NumPy استفاده شده است.

شما در گام اول باید یک Dockerfile نوشته که به کمک آن یک ایميج داکر<sup>3</sup> بر پایه پایتون بسازید که در آن امکان اجرای کد داده شده وجود داشته باشد.

همچنین لازم است که Dockerfile خود را به گونه‌ای بنویسید که پس از ایجاد کانتینر<sup>4</sup> از ایميجی که ساخته‌اید، کد داده شده بلافاصله پس از بالا آمدن کانتینر، در آن اجرا شود و نتایج آن نمایش داده شود.

\*می‌توانید از ایميج python:3.9.19-alpine که حجم کمتری دارد به عنوان ایميج پایه<sup>5</sup> استفاده نمایید.

2. با استفاده از Dockerfile ی که نوشته‌اید، ایميج خود را build نمایید.

3. ایميج ساخته شده را بر روی داکرهاب push نمایید.

4. برای تست کردن ایميج ساخته شده، ایميج خود را از داکرهاب pull کنید و يك کانتینر از آن بالا بیاورید.

موارد زیر را در فایل گزارش نمایش دهید:

- ارسال ایميج ساخته شده بر روی داکرهاب و نتیجه آن
- نمایش لیست ایميج های موجود بر روی سیستم خود
- دریافت ایميج ساخته شده از داکرهاب
- ساخت کانتینر از ایميج دریافت شده از داکرهاب
- نمایش نتیجه ساخت کانتینر

### بخش ۲

در فایل گزارش خود بیان کنید تفاوت RUN و CMD در Dockerfile چیست؟

---

<sup>3</sup> Docker Image

<sup>4</sup> Container

<sup>5</sup> Base Image

### بخش ۳)

فرض کنیم یک Dockerfile با محتویات زیر داریم:

```
1 FROM busybox:latest
2 RUN dd if=/dev/random of=/home/disk.img bs=1MB count=200
3 RUN rm /home/disk.img
```

در خط ۲ یک فایل با حجم ۲۰۰ مگابایت می‌سازیم. سپس در دستور خط ۳ این فایل را پاک می‌کنیم.

از طریق دستور زیر یک Image از روی این Dockerfile می‌سازیم و اسم آن را strange:latest می‌ذاریم:  
cat Dockerfile | docker build -t strange:latest -

انتظار داریم حجم Image ساخته شده از این Dockerfile برابر با حجم Image اولیه (busybox) باشد. در حالی که اگر حجم Image را بررسی کنیم، حجم این دو Image متفاوت است.  
حجم Image در خروجی دستور زیر نوشته شده است:

```
docker images | grep -E 'strange|busybox'
```

در این بخش از شما می‌خواهیم دلیل این تفاوت را بررسی کنید. توضیح دهید چرا چنین اتفاقی می‌افتد. سپس راه‌حل برای رفع این مشکل بیان کنید. تمامی این موارد را در فایل گزارش خود باید ذکر نمایید.

### بخش ۴)

ما برای یک پروژه‌ای Dockerfile زیر را داشتیم:

```
FROM alpine:latest
COPY ./user-pass.txt /home/user-pass.txt
RUN rm /home/user-pass.txt
```

متأسفانه تمام فایل‌های پروژه شامل فایل user-pass که حاوی username و password بوده حذف شده‌اند. تنها یک Image ساخته شده با استفاده از این Dockerfile داریم.  
در این تمرین از شما می‌خواهیم که آن username و password را برای ما پیدا کنید!

این Image را می‌توانید با استفاده از دستور زیر Pull نمایید:

```
docker pull abazshoushtari/cc-ta-hw2docker-ctf
```

مراحل انجام کار را در فایل گزارش خود توضیح و نمایش دهید.

## بخش امتیازی

برقراری اتصال ssh میان دو کانتینر ubuntu

شما باید دو کانتینر ubuntu بالا آورده که بر روی یکی از آنها پکیج openssh-server و بر روی دیگری

openssh-client نصب شده باشد. پس از آن کانفیگ های لازم را انجام دهید و در نهایت از یک کانتینر (کلاینت) به

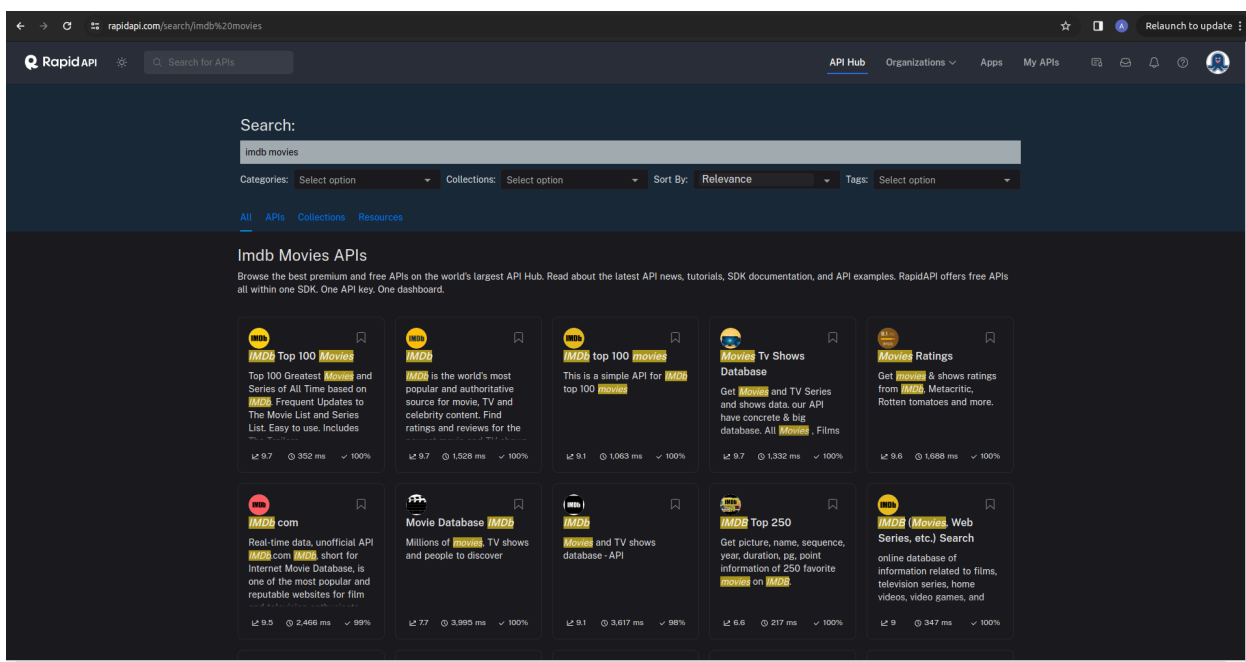
کانتینر دیگر (سرور) ssh بزنید.

مراحل انجام کار را در گزارش خود ذکر نمایید.

# فاز ۱

در این فاز از شما میخواهیم که با استفاده از ایمپچ داکر elasticsearch یک سرویس جستجوی local بالا بیاورید. همچنین از یک api به عنوان support استفاده میکنیم که در صورت عدم پیدایش document مناسب برای query وارد شده، از api استفاده میکند تا به نتیجه مرتبط برسد.

هدف ما در این تمرین زدن یک سرویس جستجوی فیلم است. دیتای تعدادی از برترین فیلمهای imdb به صورت آماده برای insert کردن در elastic در اختیار شما قرار داده میشود. همچنین میتوانید از هر api جستجوی فیلم دلخواهی استفاده کنید. پیشنهاد ما استفاده از یکی از apiهای سایت rapid api هست که در تمرین قبل هم از api دیگر این سایت استفاده کردید.



با جستجوی عبارت imdb movies با کلی api رو به رو میشوید که اکثر آنها قابلیت جستجو دارند. در انتخاب api خود مختار هستید و سختگیریای در این زمینه نخواهد شد.

همچنین از یک کانتینر redis برای cache کردن استفاده کنید.

به صورت کلی ایمپج‌های داکر مدنظر برای این تمرین مطابق زیر است:

## 1. elasticsearch

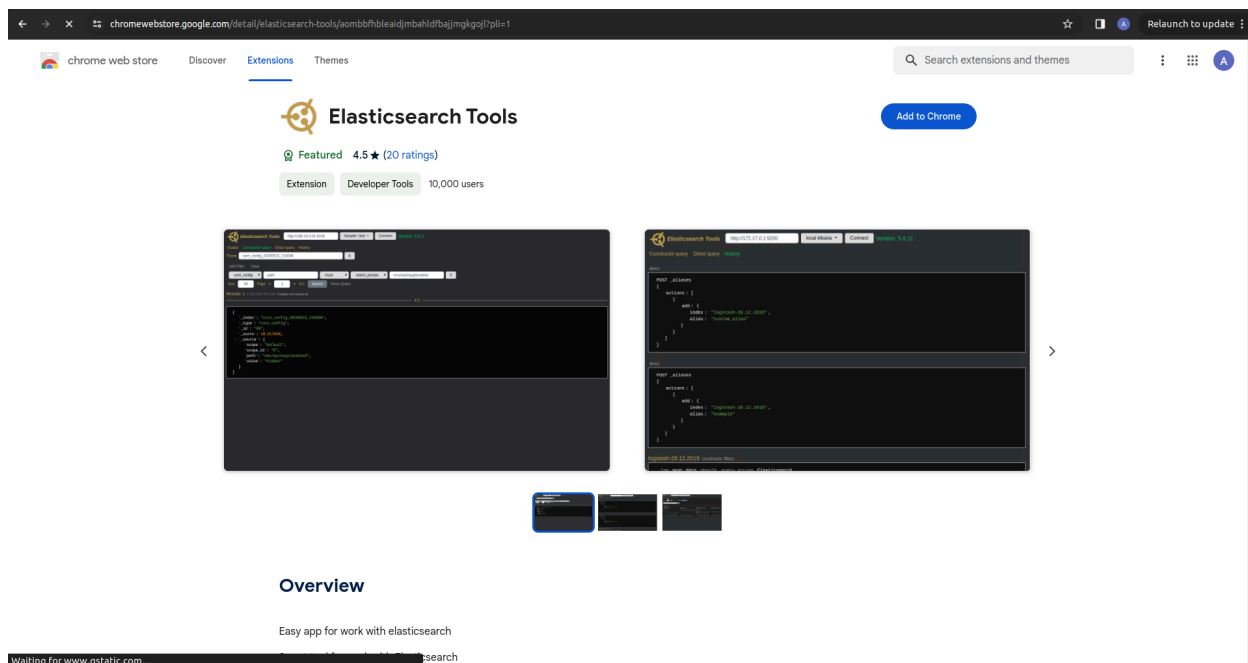
با دستور `docker pull elasticsearch:8.12.2` می‌توان ایمپج `elastic` را از `dockerhub` گرفت و از آن استفاده کرد. در هنگام `run` کردن این ایمپج در `set` کردن `flag`های ایمپج دقت کنید تا به مشکل نخورید. پس از بالا آمدن ایمپج `elastic`، نوبت `insert` کردن فایل اسامی فیلم‌های داده شده در آن است. برای این کار روش‌های گوناگونی وجود دارد که تمامی آن‌ها مورد قبول هستند. هم می‌توانید از ابزارهای آماده استفاده کنید، هم می‌توانید `extension` مربوط به `elastic` را بر روی `web browser` خود نصب کنید و از آن استفاده کنید، و هم با استفاده از دستور زیر می‌توانید این کار را انجام دهید:

```
curl -X POST "localhost:9200" -H "Content-Type: application/json" -d @document.json
```

دقت کنید که به جای `document.json` باید نام فایل خود را بگذارید، همچنین اگر در آدرسی غیر از `localhost:9200` ایمپج را `run` کردید، آدرس را هم به آدرس مد نظرتان تغییر بدهید.

دقت کنید که `elastic` یک ابزار سرچ بسیار قدرتمند با امکانات بسیار زیاد است که استفاده از این امکانات جزو اهداف آموزشی ما نیست و صرف استفاده از `query`های ابتدایی `elastic` برای کسب نمره کامل این بخش کافیهست.

همچنین برای مشاهده و بررسی بهتر اطلاعات داخل `elastic` توصیه می‌کنیم که از افزونه‌های مرورگر مناسب `elastic` استفاده کنید. مانند عکس زیر:



## 2. redis

با استفاده از redis قصد داریم تا از caching در برنامه خود استفاده کنیم. به این صورت که ابتدا در redis به دنبال پاسخ میگردیم و در صورت عدم وجود پاسخ در redis به ترتیب از elastic و api برای یافتن پاسخ استفاده می‌کنیم. دقت کنید که در صورت یافتن نتیجه باید query وارد شده و نتیجه آن را به صورت key-value در redis قرار دهید. برای دریافت ایمپج redis می‌توانید از دستور docker pull redis استفاده کنید.

## 3. your program

این ایمپج، در واقع کدی هست که شما نوشتید. کدی که باید بنویسید باید دارای یک http endpoint برای جستجو باشد که در آن ابتدا به redis، سپس elastic و سپس به api درخواست می‌فرستد. در هر مرحله اگر به جواب برسد دیگر نیازی به ارسال درخواست به بخش‌های بعدی نیست. اگر هم عبارت سرچ شده حتی پس از ارسال درخواست به api یافت نشد، پیغام مناسب را برگرداند.

## Dockerfile

برای قطعه کدی که زدید (با استفاده از هر زبان دلخواه)، یک Dockerfile بنویسید که ایمپج کد خود را بسازد.

## Docker compose

برای اجرای تمام اجزا و ایمپج‌های این برنامه، از docker compose استفاده کنید. در این فایل حتما network و volume در نظر بگیرید. برای برنامه خود در فایل docker compose آدرس Dockerfile برنامه خود را بدهید. برای اینکه elastic پس از بالا آوردن docker compose، اطلاعات فیلم‌ها را داشته باشد می‌توانید از هر روشی استفاده نمایید؛ مانند ساختن یک ایمپج با وجود از پیش اطلاعات فیلم‌ها و استفاده از آن به جای ایمپج اصلی elastic و یا استفاده از curl و یا وارد کردن آن در قطعه کد نوشته شده یا هر روش دیگری که مد نظر دارید.

برای فاز ۱ از موارد زیر اسکرین شات گرفته و در گزارش خود قرار دهید:

- نتیجه جستجوی یک کوئری که در فایل json موجود است
- نتیجه جستجوی یک کوئری که در elastic موجود نیست و در api موجود است
- نتیجه جستجوی یک کوئری که در هیچ کدام موجود نیست
- نتیجه جستجوی یک کوئری که از کش خوانده شود

## بخش امتیازی

از Kibana در کنار Elastic استفاده نمایید. (Kibana یک رابط گرافیکی است که به شما امکان استفاده از داشبورد گرافیکی را روی داده‌های داکيومنت‌های جمع‌آوری شده در Elasticsearch Cluster می‌دهد)



## نکات مربوط به تحویل تمرین

- تمرین دارای تحویل آنلاین می‌باشد. از استفاده از کدهایی که توانایی توضیح آنها را ندارید بپرهیزید!
- سوالات خود را می‌توانید با تدریس‌یاران مرتبط مطرح نمایید.
- هرگونه تقلب باعث صفر شدن طرفین می‌شود.

## مواردی که باید ارسال شوند:

یک فایل زیپ با نام **StudentID\_HW2.zip** که شامل موارد زیر می‌باشد (هر کدام از موارد را در پوشه‌های جداگانه قرار دهید)

- برای فاز ۰ موارد زیر آپلود شوند
  - Dockerfile نوشته شده برای فاز ۰ بخش ۱
  - در صورتی که برای رفع مشکل موجود در فاز ۰ بخش ۳، Dockerfile جدید نوشته‌اید، آن را نیز آپلود نمایید.
- برا فاز ۱ تمامی فایل‌های پروژه (شامل کدها و configها) به همراه Dockerfile و docker compose پروژه
- گزارشی که حداقل باید شامل موارد مطرح شده در توضیحات تمرین (به همراه اسکرین شات) باشد.

## موفق باشید

تیم تدریسیاری درس مبانی رایانش ابری