



دانشگاه صنعتی امیرکبیر  
(پلی تکنیک تهران)  
دانشکده مهندسی کامپیوتر



## رایانش ابری

(پاییز ۱۴۰۲)

تمرین دوم (بخش اول)

داکر

طراحی تمرین:

زارعی و بالی

استاد درس:

آقای دکتر جوادی

مهلت نهایی ارسال پاسخ: ۱۴۰۲/۰۸/۲۶

## مقدمه

هدف از این تمرین، کار با داکر<sup>۱</sup> است. بنابراین در این تمرین یک پروژه بسیار ساده را با استفاده از داکر کانتینرایز<sup>۲</sup> می‌کنید. برای پیش‌نیاز، لازم است داکر را بر روی سیستم خود نصب کرده باشید. برای راهنمایی می‌توانید از لینک‌های زیر استفاده کنید. دقت داشته باشید که برای اتصال به dockerhub، نیاز به تنظیم [شکن](#) یا استفاده از فیلترشکن دارید.

<https://docs.docker.com/get-docker/>

## گام اول

در ابتدا باید در [داکرها](#) یک اکانت بسازید.

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

۱. یک ایمج داکر<sup>۳</sup> بر پایه لینوکس (ترجیحا alpine ولی برای راحتی کار ubuntu هم مشکلی ندارد) بسازید که در آن امکان استفاده از دستور curl وجود داشته باشد.

۲. ایمج ساخته شده را بر روی داکرها آپلود نمایید.

۳. برای تست کردن ایمج ساخته شده، ایمج خود را از داکرها دانلود کنید و یک کانتینر از آن بالا بیاورید.

۴. یک درخواست curl به google.com ارسال کنید.

موارد زیر را در فایل گزارش نمایش دهید:

- ارسال ایمج ساخته شده بر روی داکرها و نتیجه آن
- نمایش لیست ایمج‌های موجود بر روی سیستم خود
- دریافت ایمج ساخته شده از داکرها
- ساختن کانتینر از ایمج دریافت شده از داکرها
- اجرا دستور curl و نتیجه آن

## گام دوم

در این گام قرار است که سروری را برای دریافت اطلاعات مربوط به آب و هوا را توسعه دهیم و به صورت مشخص، دمای شهرها را نگهداری کنیم و برگردانیم. محدودیتی در API استفاده شده وجود ندارد و بسته به سلیقه خود می‌توانید از هر API موجود برای دریافت اطلاعات استفاده کنید. در قسمت زیر لیست API‌های پیشنهادی آورده شده‌اند:

---

<sup>1</sup> Docker

<sup>2</sup> Containerize

<sup>3</sup> Docker Image

- [WeatherAPI.com](https://weatherapi.com)
- [Ninjas](https://ninjas.p.rapidapi.com)

در این بخش لازم است که به ازای هر شهر دو مقدار را ذخیره نمایید. در صورت استفاده از WeatherAPI، به ازای هر شهر دما به سانتیگراد و فارنهایت و در صورت استفاده از Ninjas، اطلاعات مربوط به بیشینه و کمینه دما را به ازای هر شهر ذخیره نمایید.

**توجه:** برای استفاده از API های ذکر شده، نیاز است در سایت مربوطه ثبت نام کرده و در قسمت pricing برنامه، رایگان را subscribe کنید تا API key مخصوص به آن را بسازید و به عنوان هدر به درخواست HTTP خود اضافه کنید.

**توجه:** استفاده از زبان برنامه نویسی دلخواه برای توسعه پروژه آزاد است.

نمونه هایی از درخواست های ارسال شده با api های پیشنهادی:

```
curl --request GET \
  --url 'https://weatherapi-com.p.rapidapi.com/current.json?q=35.6895%20%2C51.3890' \
  --header 'X-RapidAPI-Host: weatherapi-com.p.rapidapi.com' \
  --header 'X-RapidAPI-Key: dc743a6ccemshaf4d7ccbc56663ep15e0b0jsn12a25a49e0e2'
```

```
▼ location: {} 8 keys
  name: "Tehran"
  region: "Tehran"
  country: "Iran"
  lat: 35.69
  lon: 51.39
  tz_id: "Asia/Tehran"
  localtime_epoch: 1695660528
  localtime: "2023-09-25 20:18"
▼ current: {} 23 keys
  last_updated_epoch: 1695660300
  last_updated: "2023-09-25 20:15"
  temp_c: 25
  temp_f: 77
  is_day: 0
```

در این api از longitude و latitude شهر تهران استفاده شده است و دما ۲۵ درجه سانتی گراد و ۷۷ درجه فارنهایت، نمایش داده شده است.

```
curl --request GET \
  --url 'https://weather-by-api-ninjas.p.rapidapi.com/v1/weather?city=Tehran' \
  --header 'X-RapidAPI-Host: weather-by-api-ninjas.p.rapidapi.com' \
  --header 'X-RapidAPI-Key: dc743a6ccemshaf4d7ccbc56663ep15e0b0jsn12a25a49e0e2'
```

```
▼ {} 10 keys
  cloud_pct: 20
  temp: 25
  feels_like: 24
  humidity: 13
  min_temp: 25
  max_temp: 25
  wind_speed: 3.09
  wind_degrees: 130
  sunrise: 1695608642
  sunset: 1695652100
```

در این api از متغیر city برای مشخص کردن شهر مورد نظر استفاده شده است و کمینه دما ۲۵ درجه و بیشینه دما ۲۵ درجه سانتی گراد نمایش داده شده است.

همچنین برای جلوگیری از ارسال درخواست‌های اضافه در برنامه خود، باید از کش [Redis](#) استفاده کنید. برای استفاده از ردیس، ایمپج آن را دانلود کنید و یک کانتینر از آن بالا بیاورید. در کلید اسم شهر یا مختصات شهر مورد نظر و در مقدار دمای آن شهر را نگه دارید و مقادیر باید به صورت پیش فرض به مدت پنج دقیقه اعتبار داشته باشند.

حال باید برای برنامه خود یک Dockerfile بنویسید و آن را build کنید و ایمپج خود را بسازید و بر روی داکرهاب قرار دهید.

توجه: در این مرحله شما باید دو کانتینر داشته باشید. یکی مربوط به برنامه خود و دیگری مربوط به کش ردیس.

**توجه:** برای برقراری ارتباط با کش ردیس از داخل برنامه خود، یک network بین این دو کانتینر تعریف کنید تا کانتینرها بتوانند همدیگر را ببینند و با هم در ارتباط باشند.

می‌دانیم با پایین آمدن کانتینر ردیس، اطلاعات از دست خواهند رفت. برای جلوگیری از این اتفاق، باید اطلاعات کش ردیس را persist کنیم تا با پایین آمدن آن، اطلاعات از بین نروند و پس از بالا آمدن مجدد، اطلاعات بازیابی شوند. برای این منظور باید از قابلیت Volume استفاده کنید.

در نهایت فایل [داکر کامپوزی](#) تعریف نمایید که ایمپج‌های مربوطه درگام‌های قبل درون آن تعریف شود؛ همچنین لازم است که **network** و **volume** مربوطه نیز در این فایل تعریف شود. توجه شود که در این فایل باید از نسخه داکر فایل‌هایی که نوشته‌اید **build** بگیرید.

**نکته مهم:** پروژه باید برای فیلدهای زیر کانفیگ پذیر باشد (به این معنی که به صورت hard code نباشند):

- شماره پورتی که بر روی آن سرور خود را بالا می‌آورید

- مدت زمان اعتبار کلیدها در کش ردیس
- نام شهری یا مختصاتی که دمای آن را می‌خواهیم
- API key

موارد زیر در فایل گزارش نمایش داده شود:

- دریافت ایمپج ردیس و ساختن کانتینر از آن
- ساختن شبکه برای برقراری ارتباط بین دو کانتینر
- ساختن Volume جهت persist کردن اطلاعات کش ردیس
- ساختن ایمپج سرور نوشته شده با استفاده از داکر فایل
- ارسال ایمپج ساخته شده بر روی داکرهاب و نمایش نتیجه آن
- نمایش اطلاعات ایمپج سرور خود با استفاده از دستور `docker inspect`
- نمایش کانتینرهای موجود در سیستم خود با استفاده از دستور `docker ps`
- نمایش میزان منابع استفاده شده توسط کانتینرهای موجود با استفاده از دستور `docker stats`

## نکات مربوط به تحویل تمرین

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

## مواردی که باید ارسال شود

یک فایل زیپ با نام SID\_HW2.zip مانند 9831031\_HW2.zip که شامل موارد زیر است (هر مورد را در فولدر جداگانه قرار دهید)

- برای گام اول لازم است موارد زیر آپلود شود:
  - Dockerfile نوشته شده برای ساخت ایمیج مورد نظر
  - تمامی دستوراتی که برای ساخت ایمیج مورد نظر از طریق docker commit انجام داده‌اید
- برای گام دوم تمامی فایل‌های پروژه به همراه Dockerfile، فایل config آن و داکر کامپوز پروژه
- گزارشی که حداقل باید شامل موارد مطرح شده در توضیحات تمرین (به همراه اسکرین‌شات) باشد

موفق باشید

تیم تدریس‌یاری درس مبانی رایانش ابری