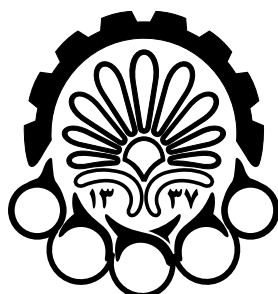


به نام خدا



دانشگاه صنعتی امیرکبیر  
( پلی تکنیک تهران )

رایانش ابری

## تمرین دوم داکر و مقدمات کوبرنتیز

طراحی تمرین:

آقایان احمدوند، فرنگی زاده و حسینی

استاد درس:

آقای دکتر جوادی

مهلت نهایی ارسال پاسخ:

۱۹ فروردین ماه ۱۴۰۱ ساعت ۲۳:۵۹

نکته مهم: دقت کنید که تمدید نخواهیم داشت و صرفاً می‌توانید ۷ روز از ۱۴ روز مجاز برای تاخیر ارسال تمرین‌ها در این ترم را استفاده کنید. به ازای هر روز تاخیر مازاد تا زمان ارائه اسکایپی، ده درصد نمره آن تمرین به عنوان جریمه، از نمره نهائی تمرین کسر می‌شود. ما در اعمال این قاعده جدی هستیم.

## مقدمه

هدف از این تمرین کار با داکر و پیدا کردن شناخت اولیه با کوبرنتیز است. لذا قصد داریم یک پروژه بسیار ساده را با استفاده از داکر containerize کرده و بر روی کوبرنتیز دیپلوی کنیم. برای انجام این تمرین لازم است docker و minikube را بر روی سیستم خود نصب کرده باشد. به این منظور می‌توانید از لینک‌های زیر کمک بگیرید:

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

<https://minikube.sigs.k8s.io/docs/start/>

## گام اول

در قدم اول شما باید یک ایمج<sup>۱</sup> داکر 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. سپس ایمج ساخته شده را بر روی داکرهاب آپلود کنید. در نهایت به منظور تست کردن آن، با کمک دستور docker run ایمج را از داکرهاب خود دریافت کرده و یک کانتینر از آن بالا بیاورید. حال با ارسال یک درخواست curl به google.com خروجی دریافت کنید.

موارد زیر را در فایل گزارش نمایش دهید:  
(۱) ارسال ایمج ساخته شده بر روی داکرهاب و نتیجه آن

```
docker image push <dockerhub username>/<image name>
```

(۲) دریافت و اجرای ایمج ساخته شده از داکرهاب

```
docker run -it <dockerhub username>/<imagename>
```

(۳) اجرا و خروجی دستور curl



```
saman — docker run -it --rm net-utils:1.0 — docker — com.docker.cli • docker run -it --rm net-utils:1.0 — 80x24
[~] 23:42:45
[ docker run -it --rm net-utils:1.0 ]
bash-5.1# curl google.com
<HTML><HEAD><meta http-equiv="content-type" content="text/html; charset=utf-8">
<TITLE>301 Moved</TITLE></HEAD><BODY>
<H1>301 Moved</H1>
The document has moved
<A HREF="http://www.google.com/">here</A> .
</BODY></HTML>
bash-5.1#
```

---

<sup>1</sup> image

## گام دوم

در این گام قصد داریم یک سرور بسیار ساده توسعه دهیم که با ارسال درخواست به آن، اسم host و زمان را برای ما ارسال کند:

```
{
  "hostname":      my-user@my-host,
  "temperature":  temperature,
  "weather_descriptions": weather_descriptions,
  "wind_speed":    wind_speed,
  "humidity":      humidity,
  "feelslike":     feelslike,
}
```

فیلد اول توسط خود پروژه مقدار دهی می‌شود. اما برای فیلدهای بعدی که مربوط به وضعیت آب‌وهوا است، شما باید به آدرسی که در یک فایل کانفیگ مشخص می‌شود درخواست GET ارسال کرده و فیلدهای مورد نیاز را از پاسخ استخراج کرده، در این فیلد قرار دهید.

```
└─ curl --request GET \
  --url 'http://api.weatherstack.com/current?access_key=efdcd00c3001f0000fa7507c0b97f85d&query=Tehran'
{"request":{"type":"City","query":"Tehran, Iran","language":"en","unit":"m"},"location":{"name":"Tehran","country":"Iran","region":"Tehran","lat":"35.726","lon":"51.330","timezone_id":"Asia/Tehran","localtime":"2022-03-03 11:25","localtime_epoch":1646306700,"utc_offset":"+3.50"},"current":{"observation_time":"07:55 AM","temperature":17,"weather_code":116,"weather_icons":["https://assets.weatherstack.com/images/wsymbols01_png_64/wsymbol_0002_sunny_intervals.png"],"weather_descriptions":["Partly cloudy"],"wind_speed":11,"wind_degree":190,"wind_dir":"S","pressure":1017,"precip":0,"humidity":29,"cloudcover":75,"feelslike":17,"uv_index":4,"visibility":10,"is_day":"yes"}}}
```

پروژه شما می‌تواند به هر زبانی توسعه داده شود. پس از اتمام پیاده‌سازی، برای پروژه خود یک Dockerfile بنویسید که با استفاده از آن، بتوان پروژه را containerize کرد. در نهایت با build کردن Dockerfile ایمج پروژه خود را تولید کرده و بر روی داکرهاب قرار دهید.

نکته مهم: پروژه شما باید برای port سرور و آدرسی که جهت ارسال درخواست آب‌وهوا استفاده می‌شود، کانفیگ پذیر باشد و این مقادیر نباید داخل کد شما hard code شده باشند. (می‌توانید از آدرس قابل مشاهده در عکس بالا به عنوان آدرس api دیفالت استفاده کنید. پورت دیفالت پیشنهادی 8080 است)

موارد زیر را در فایل گزارش نمایش دهید:  
(۱) build کردن ایمج با استفاده از Dockerfile ساخته شده

`docker build -t <image name>.`

(۲) ارسال ایمج ساخته شده بر روی داکرهاب و نتیجه آن

`docker tag <image name> <dockerhub username>/<image name>`

`docker image push <dockerhub username>/<image name>`

(۳) در صورتی که پروژه خود را با استفاده از ایمج ساخته شده بر روی سیستم شخصی خود تست کردید، تصاویر مربوطه را قرار دهید (این مرحله اجباری نیست ولی توصیه می‌شود)

## گام سوم

حال زمان این است که با نوشتن فایل‌های دیپلویمنت کوبرنتیز، پروژه خود را بر روی minikube بالا بیاوریم.

اولین کامپوننت مورد نیاز یک ConfigMap برای پروژه است تا بتوان port سرور و آدرس api مربوط به آب‌وهوا از آن خوانده شود. کامپوننت بعدی یک Deployment است که وظیفه آماده سازی و نگهداری podها را بر عهده دارد. تعداد replica را برابر با ۲ تعیین کنید. آخرین مورد یک Service است که با استفاده از آن می‌توانیم به پروژه و در واقع سروری که توسعه داده‌ایم دسترسی داشته باشیم. پس از ساخت فایل‌های گفته شده، آن‌ها را به همان ترتیب و با استفاده از دستور kubectl apply بر روی کلاستر minikube ایجاد کنید.

موارد زیر را در فایل گزارش نمایش دهید:

(۱) با استفاده از دستور kubectl get صحت ایجاد منابع بر روی کلاستر را نمایش دهید

(۲) آدرس IP پادها و نحوه برقراری ارتباط میان آن‌ها و سرویس ساخته شده

## گام چهارم (آخر)

در قدم آخر می‌خواهیم سیستم را تست کنیم. به این منظور ایمج ساخته شده در گام اول را با کمک دستور kubectl run بر روی کلاستر اجرا کنید و به سرویس ساخته شده برای پروژه درخواست بزنید.

موارد زیر را در فایل گزارش نمایش دهید:

(۱) با استفاده از دستور kubectl get صحت ایجاد پاد بر روی کلاستر را نمایش دهید

(۲) چند مورد از درخواست‌های ارسال شده به پروژه خود را همراه با توزیع بار میان پادها نشان دهید

(۳) دستور مورد استفاده برای اجرا کردن ایمج گام اول

## نکات مربوط تحویل به پروژه

- پروژه شما تحویل اسکایی خواهد داشت بنابراین از استفاده از کدهای یکدیگر یا کدهای موجود در وب که قادر به توضیح دادن عملکرد آنها نیستید، بپرهیزید.
- در تحویل اسکایی از شما خواسته می‌شود تا با استفاده از فایل‌های دیپلویمنت نوشته شده، پروژه خود را از صفر بر روی کلاستر کوبرنتیز ببرید و در تعداد پادهای پروژه تغییر ایجاد کنید. همچنین باید بلد باشید پس از تغییر فایل کانفیگ آن را بر روی پادها اعمال کنید.
- ابهامات خود را در سایت و یا گروه تلگرامی درس مطرح کنید و ما در سریعترین زمان ممکن به آنها پاسخ خواهیم داد.

## آنچه که باید ارسال کنید

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

- برای گام اول لازم است یکی از موارد زیر آپلود شود
  - (۱) Dockerfile نوشته شده برای ساخت ایمج مورد نظر (پیشنهاد ما)
  - (۲) تمامی دستوراتی که برای ساخت ایمج مورد نظر از طریق docker commit انجام داده‌اید
- برای گام دوم تمامی فایل‌های پروژه به همراه Dockerfile و فایل config آن
- برای گام سوم ۳ فایل دیپلویمنت ساخته شده
- گزارش که حداقل باید شامل موارد مطرح شده در توضیحات تمرین باشد.

موفق باشید

تیم درس مبانی رایانش ابری