

دوره آموزشی داکر

پیش زمینه و هدف

فرض کنید می‌خواهیم محیطی که برای یک پروژه موقع توسعه روی سیستم خودمون درست کردیم (مثل نصب و کانفیگ دیتابیس و nginx) رو یک‌جای دیگه (مثل پروداکشن یا سیستم یکی دیگه از توسعه‌دهنده‌ها) درست کنیم. درست کردن این محیط میتونه کار خیلی سخت و طاقت‌فرسایی باشه به همین دلیل داکر به کمک ما میاد تا این کار رو راحت‌تر کنه. در واقع به کمک داکر می‌تونیم محیطی که برنامه قراره توش اجرا بشه رو در کنار کدهای برنامه نگه‌داری کنیم و هروقت و هرکجا که خواستیم به سادگی اون محیط رو ایجاد کنیم و برنامه‌مون رو اجرا کنیم.

مقدمات

قبل از هرچیز خوبه اول با مفاهیم مجازی‌سازی و به خصوص مجازی‌سازی در سطح سیستم‌عامل آشنا بشیم.

مطالعه

برای آشنایی با مفاهیم مجازی‌سازی [این](#) ویدیو رو ببینید.

مطالعه بیشتر

برای آشنایی با مفاهیم Containerization و تفاوتش با Virtualization [این مقاله](#) رو بخونید و [این](#)

[ویدیو](#) رو ببینید.

نصب داکر

بسته به روشی که داکر رو نصب می کنید احتمالاً در این قسمت نیاز دارید VPN سیستم روشن باشد.

مطالعه

به کمک این [مقاله](#) داکر رو روی سیستمتون نصب کنید.

برای اینکه مطمئن شویم داکر به درستی روی سیستم نصب شده دستور زیر رو اجرا کنید. در صورتی که خروجی چیزی مشابه زیر باشه یعنی داکر درست نصب شده.

```
$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
NAMES					

مطالعه

اگر موقع اجرای دستور با خطای permission denied مواجه شدید یا باید موقع کار با داکر از sudo استفاده کنید یا به کمک این [مقاله](#) مشکل رو حل کنید.

بزن بریم

تا اینجا با مفهوم کانتینر آشنا شدیم. حالا میخوایم ببینیم چطور میتونیم یک کانتینر بسازیم. توی داکر کانتینرها از روی ایمیجها ساخته میشن. در واقع اگه بخوایم به زبانهای شی گرا تشبیه کنیم ایمیجها معادل کلاسها و کانتینرها متناظر با آبجکتها میشن.

مطالعه

بخش‌های ۲ تا ۹ (به جز ۴) از راهنمای شروع داکر رو از [اینجا](#) مطالعه کن و تمرین‌هاش رو انجام بده. (تاکید بیشتر رو استفاده از کامندهای cli هست و اگه توی کار با پنل وبش به مشکل خوردی بپر بره!) همینطور خوبه مفاهیم شبکه‌ی داکر رو از [اینجا](#) به نگاه بندازی و [پایلیش شدن پورت](#) هم بدونی چیه.

اگر موقع دانلود ایمج از مخازن **dockerhub** به مشکل تحریم یا **ratelimit** خوردید میتونید به جای VPN از **mirror repository** استفاده کنید.

```
$ docker pull nginx:latest
```

```
Error response from daemon: toomanyrequests: You have reached your  
pull rate limit. You may increase the limit by authenticating and  
upgrading: https://www.docker.com/increase-rate-limit
```

```
$ docker pull docker.repos.balad.ir/nginx:latest
```

```
Pulling...
```

```
$ docker run docker.repos.balad.ir/nginx:latest
```

Multistage Docker Builds

تا اینجا مفاهیم اولیه داکر رو یاد گرفتیم و الان میتونیم برای پروژمون Dockerfile بنویسیم. حالا یه پروژه فرضی رو در نظر بگیرید که به زبان Go نوشته شده که میخوایم براش Dockerfile بنویسیم. یه Dockerfile نمونه میتونه چیزی مشابه زیر باشه.

```
FROM golang:1.16
WORKDIR /source
ADD . .
RUN go build -o /run.out
CMD [ "/run.out" ]
```

با وجود اینکه Dockerfile بالا ایمیج درستی رو برامون تولید میکنه اما یه مشکل خیلی مهم داره و اونم اینه که حجم ایمیج تولید شده خیلی زیاده. آیا واقعا نیازه که توی ایمیج نهایی که ساخته میشه کامپایلر Go نصب باشه؟!

برای حل این مشکل میشه Dockerfile رو اصطلاحا به صورت چندمرحله‌ای (Multistage) نوشت یا حتی به فایل‌های مختلف شکوند.

مطالعه

برای یادگیری Multistage Docker Builds این [مقاله](#) رو بخونید.

مطالعه بیشتر

یکی از پرکتیس‌های خوب برای بیلد کردن داکر استفاده از buildkit هست که خیلی ساده است. [اینجا](#) میتونی در موردش بیشتر بخونی و استفاده کنی.

تمرین

توی این تمرین قراره با استفاده از docker-compose یک دیتابیس postgresql بیاریم بالا و یه کد پایتون رو که از قبل نوشتیم، dockerize کنیم که بتونه وصل بشه به دیتابیس و کوئری بزنه

۱. با استفاده از docker-compose یک دیتابیس postgresql بیار بالا و با استفاده از volume دیتاش رو پرسیست کن. بعد با استفاده از docker exec، موارد زیر رو داخل دیتابیس بساز:

- یک دیتابیس به اسم mentorship
- یک یوزر جدید به اسم mentor و پسورد دلخواه که به دیتابیس mentorship دسترسی داشته باشه
- یک تیبل داخل دیتابیس mentorship به اسم information که دو تا ستون داره. یکی آیدی و دیگری name.

دقت کن که برای بالا آوردن postgres با داکر کامپوز لازم هست یک سری متغیرها رو (مثل پسورد دیفالت و یوزرنیم دیفالت) از طریق environment به سرویس پستگرس پاس بدی. این اطلاعات، غیر از ۳ موردی هست که بالاتر ازتون خواسته شده بسازی.

۲. فایل [db_worker.py](#) رو نگاه کن. توی این فایل یه اپلیکیشن ساده فلسک پیاده شده که به دیتابیس وصل میشه و ۲ تا api ساده داره برای insert و select. برای این فایل، requiremets.txt بساز و براش Dockerfile بنویس. همین طور به docker-compose هم اضافه‌ش کن تا بتونه به دیتابیسی که توی مرحله ۱ درست کردی وصل بشه و کار کنه

در نهایت برای اینکه مطمئن بشی همه چیز اوکیه میتونی به این آدرس‌ها ریکوئست بزنی:

localhost:5000/insert/mojtaba

localhost:5000/select/mojtaba

حالا كه اين بخش رو تموم كردى ممنون مى شم كه فرم [يا زخورد](#) رو پر كنى. 