

پروژه رایانش ابری

فاز سه

استاد درس: دکتر ایزدپناه

مقدمه

پس از آشنایی با docker و کار با کانتینر ها لازم است تا با روشی بهینه تر برای اجرای چندین container با هم آشنا شویم. ابزاری که برای این کار میتوان از آن استفاده کرد docker-compose است. برای یادگیری نحوه کار با docker-compose ویدیوهای آموزشی و منابع لازم در اختیار شما قرار میگیرد.

مواردی که در این پروژه انتظار می رود یاد گرفته شوند :

- آشنایی با docker-compose
- آشنایی با فایل های yml یا yml
- آشنایی با دستورات docker-compose

در این فاز نیاز است شما یک فایل داکر کومپوز برای اپلیکیشنی که در اختیار شما قرار می گیرد بنویسید.

توضیحات لازم در مورد اپلیکیشن:

: Simple-go-rest-api

این اپلیکیشن که به زبان go نوشته شده است از یک دیتابیس برای ذخیره سازی متن استفاده میکند. در این اپلیکیشن endpoint های زیر تعریف شده است:

- <http://localhost:8000/healthcheck> GET
- <http://localhost:8000/texts/> POST
- <http://localhost:8000/texts/> GET
- <http://localhost:8000/texts/id> GET
- <http://localhost:8000/texts/id> DELETE
- <http://localhost:8000/texts/id> PUT

یک postman collection در اختیار شما قرار گرفته که می توانید به کمک آن در خواست های مورد نیاز را بزنید. توجه کنید که اپلیکیشن برای اجرا به دیتابیس نیاز دارد.

موارد لازم برای نوشتن فایل Docker-Compose:

دیتابیس :

برای دیتابیس یک سرویس(کانتینر) تعریف کنید و از image زیر استفاده کنید:

```
mysql/mysql-server:5.7
```

به کمک volume داده های دیتابیس را persist کنید به صورتی که با از بین رفتن یا به وجود آمدن کانتینر داده های پیش از حذف شدن کانتینر موجود باشند.

تمام environment variable های زیر را برای دیتابیس مشخص کنید:

- MYSQL_ROOT_PASSWORD
- MYSQL_USER
- MYSQL_PASSWORD
- MYSQL_DATABASE

مقدار لازم برای تمام این environment variable ها در فایل env. قرار گرفته که برای دسترسی به مقدار آنها به صورت زیر عمل کنید :

```
${MYSQL_ROOT_PASSWORD}
```

به ترتیب برای environment variable های بالا از

- MYSQL_ROOT_PASSWORD
- DB_USER
- DB_PASSWORD
- DB_NAME

استفاده کنید که مقدار آنها در env. مشخص شده است.

اپلیکیشن :

لازم است تا سرویس (کانتینر) اپلیکیشن از روی image ای که در دایرکتوری اپلیکیشن است ایجاد شود یعنی در فایل yml داکر کومپوز باید این ایمیج build شود.

پورتهای که برای اپلیکیشن مشخص شده 8000 است که باید مپ شود.

باید محلی که اپلیکیشن روی هاست قرار دارد به کمک یک volume به مسیر /app در کانتینر mount شود که با هر تغییری در کد های اپلیکیشن روی هاست این تغییرات در کانتینر قابل مشاهده باشد تا به صورت همزمان تغییرات در اپلیکیشن در حال اجرا اعمال شود.

کانتینر اپلیکیشن تا زمانی که کانتینر دیتابیس اجرا نشده است اجرا نشود.

برای کانتینر اپلیکیشن استفاده از cpu و ram به صورت دلخواه محدود شود.

نکات قابل توجه :

ممکن است در اولین اجرای دستور `docker-compose up` با ارور زیر مواجه شوید که با دوباره اجرا کردن دستور برطرف خواهد شد.

```
panic: Failed to connect to database!
```

نحوه تحویل:

فایل اپلیکیشن در `vu` قرار داده شده است پس از دریافت آن در کنار `dockerfile` یک فایل به نام `docker-compose.yml` ایجاد کنید و موارد خواسته شده را پیاده کنید.

پس پیاده سازی فایل `docker-compose.yml` و `screenshot` های خواسته شده را پس از فشرده سازی با نام `studentNum1-studentNum2-phase3.zip` در سامانه `vu` آپلود کنید.

تصویر از خروجی اجرای دستورات زیر را پس از اجرا دستور `docker-compose up` قرار دهید.

```
docker ps -a
```

```
docker images
```

و همینطور تصویر از خروجی اجرا دستور زیر برای کانتینر های دیتابیس و اپلیکیشن

```
docker container inspect CONTAINER [CONTAINER...]
```