

پروژه رایانش ابری

فاز دو

استاد درس: دکتر ایزدپناه

مقدمه

پس از آشنایی با مفاهیم containerization نیاز است با ابزار مربوطه آشنا شویم مهمترین ابزار docker است.

برای نصب و یادگیری نحوه کار با docker ویدیو های آموزشی و منابع لازم در اختیار شما قرار میگیرد.

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

- آشنایی با docker
- نصب docker
- آشنایی با دستورات docker cli
- آشنایی با docker image
- آشنایی با docker container
- آشنایی با dockerfile
- نحوه نوشتن dockerfile
- نحوه ایجاد کردن image
- نحوه ایجاد و اجرا container
- آشنایی با دستورات مربوط به image ها و container ها
- آشنایی با web server
- آشنایی و نحوه کار با postman

در این فاز نیاز است شما داکر فایل های لازم برای هر قسمت را بنویسید و image های مورد نظر را بسازید سپس با اجرای کانتینر ها کارایی آنها را چک کنید، تمام موارد لازم و دستورات مورد نیاز مرتبط با هر اپ در اختیار شما قرار میگیرد.

داکر فایل هایی که باید نوشته شوند:

- داکر فایل برای اپلیکیشن ping-pong-java
- داکر فایل برای اپلیکیشن ping-pong-python
- داکر فایل برای اپلیکیشن ping-pong-js
- داکر فایل برای nginx

توضیحات لازم برای هر قسمت:

داکر فایل برای اپلیکیشن ping-pong-java:

این اپلیکیشن شامل endpoint های زیر است:

- /ping

/ping: GET method to receive pong as response

این اپلیکیشن از پورت 8001 استفاده میکند در نتیجه url مورد نظر برابر:

localhost/8001/ping

است.

برای اجرای این اپلیکیشن از دستور زیر استفاده کنید:

```
Javac PingPong.java  
Java PingPong
```

بهتر است از openjdk:8 به عنوان image پایه استفاده شود.

توجه داشته باشید network alias این کانتینر برابر javaapp است.

داکر فایل برای اپلیکیشن ping-pong-python:

این اپلیکیشن شامل endpoint های زیر است:

- /ping
- /echo

/ping: GET method to receive pong as response

/echo: POST method with "echo":"text" to receive "text" as response

این اپلیکیشن از پورت 8002 استفاده میکند در نتیجه url مورد نظر برابر:

localhost/8002/ping

localhost/8002/echo

است.

برای اجرای این اپلیکیشن از دستور زیر استفاده کنید:

```
Uvicorn PingPong:app --host 0.0.0.0 --port 8002 --reload
```

بهتر است از tiangolo/uvicorn-gunicorn-fastapi به عنوان image پایه استفاده شود.

توجه داشته باشید در صورتی که WORDKDIR مشخص نشود اجرا با خطا مواجه خواهد شد.

توجه داشته باشید network alias این کانتینر برابر pythonapp است.

داکر فایل برای اپلیکیشن ping-pong-js:

- /ping

/ping: get method to receive pong as response

این اپلیکیشن از پورت 8003 استفاده میکند در نتیجه url مورد نظر برابر:

localhost/8003/ping

است.

برای اجرای این اپلیکیشن از دستور زیر استفاده کنید:

```
Node PingPong.js
```

بهتر است از node:slim به عنوان image پایه استفاده شود. توجه داشته باشید لازم است تمام فایل های موجود در پوشه به کانتینر انتقال داده شود. توجه داشته باشید network alias این کانتینر برابر jsapp است.

داکر فایل برای nginx:

کانفیگ مربوط برای nginx به همراه یک منبع برای یادگیری config قرار داده شده است. کانفیگ نوشته شده وظایف زیر را دارد:

- به عنوان reverse proxy برای اپلیکیشن ها عمل کند
- یک صفحه html استاتیک را میزبانی کند
- محتویات یک پوشه خارج از کانتینر را برای دانلود میزبانی کند

مسیری که nginx config در آن قرار دارد که باید با کانفیگی که در اختیار شما قرار گرفته جایگزین شود:

```
/etc/nginx/nginx.conf
```

راهنمایی:

توجه داشته باشید برای قابل دسترس بودن کانتینر برای کانتینر های دیگر نیاز است که آنها در یک network باشند.

نحوه تحویل:

پس از دریافت فایل پروژه در هر دایرکتوری مشخص شده `dockerfile` های خواسته شده را قرار دهید و در پوشه `screenshot` تصویر از خروجی اجرای دستورات زیر را پس از تست کارایی کانتینر ها قرار دهید.

```
docker ps -a
```

```
docker images
```

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

علاوه بر تصاویر بالا تصاویر جواب درخواست به هر کدام از `endpoint` های مشخص شده در کانفیگ `nginx` برای اپلیکیشن ها را به صورتی که `url` مشخص باشد نیز قرار دهید. سپس پس از فشرده سازی با نام `studentNum1-studentNum2-phase2.zip` در `vu` اپلود کنید.

منابع:

- [nginx config](#)
- [web server](#)
- [Postman](#)
- [Dockerfile](#)
- [Docker get started](#)
- [Reverse proxy](#)