

پخش کن

حتماً با اپلیکیشن مزخرفی به نام واتس‌آپ کار کرده‌اید. واتس‌آپ در تیم مزخرف خود، به همکار خوبی مثل شما نیاز دارد تا به عنوان یک مهندس، بتوانید این اپلیکیشن را بهبود دهید. یکی از مهم‌ترین کارهایی که برای نگهداری و کارایی واتس‌آپ نیاز است، ایجاد چندین سرور و پخش نمودن درخواست‌های کاربران میان آن‌هاست.

برای دانلود پروژه‌ی اولیه روی [این لینک](#) کلیک کنید.

سیستم *api*ها

واتس‌آپ، از قبل ۴ کانتینر برای *api*های خود تدارک دیده که شما از جزئیات *api*ها کاملاً بی‌خبر هستید و محتویات این *api*ها کاملاً محرمانه می‌باشد. این *api*ها، در فایل `docker-compose.yml` وجود دارند و تنها کاری که شما باید انجام دهید، اضافه کردن یک سیستم لودبالانس بر روی پورت 80 این سیستم، در کنار بقیه‌ی کانتینرها می‌باشد.

لودبالانس

اضافه کردن یک *container* در فایل `docker-compose.yml` که بر روی پورت 80 سرور فعال باشد و میان ۴ سرویس موجود، عملیات لودبالانس را به صورت کاملاً مساوی انجام دهد. به این نکته توجه داشته باشید که این لودبالانس میان سرویس‌ها باید کاملاً مساوی اتفاق بیوفتد و تمامی ریکوئست‌ها دقیقاً میان سرویس‌ها به طور برابر پخش شوند.

فراموش نکنید که تنها می‌توانید از ایمیج‌های *Standard* در رجیستری کوئرا استفاده نمایید. فرقی نمی‌کند با چه ایمیج یا تکنولوژی‌ای این کار را انجام دهید. حتی می‌توانید خودتان برنامه‌ای بنویسید که لودبالانس انجام دهد. تنها نکته‌ی مهم این است که بر روی پورت 80 آن برنامه را اجرا نمایید.

همچنین توجه داشته باشید ممکن است برچسب ایمیج موردنظر شما لزوماً `latest`: نباشد، برای مثال ایمیج `HAProxy` تنها به دو صورت `registry.gitlab.com/qio/standard/haproxy:2.5` و `registry.gitlab.com/qio/standard/haproxy:2.5-alpine` وجود دارند.

نکته‌ی مهم

شما اجازه‌ی *build* کردن ایمج جدید نخواهید داشت و در صورتی که می‌خواهید خودتان برنامه‌ای بنویسید که کار لودبالانس را انجام دهد، باید از پوشه‌ی `data` که برای ایجاد *volume* در نظر گرفتیم استفاده کنید و برنامه‌ی خود را به درون کانتینرهای موجود *volume* کنید. مثلاً برنامه‌ای پایتونی نوشته‌اید که در ایمج پایتون استاندارد کوئرا *volume* می‌شود و در *entrypoint* آن را اجرا می‌کنید.

ایجاد Volume

در صورتی‌که نیاز دارید فایل یا مسیری را به درون هریک از *Volume container* کنید، می‌توانید از پوشه‌ای به نام `data` استفاده کنید. و برای ایجاد *Volume* از *Relative Path* استفاده کنید. برای مثال:

```
services:
  sample_container:
    image: "registry.gitlab.com/qio/standard/someimage:latest"
    volumes:
      - ./data/somefile:/somefile
```

فراموش نکنید که فایل `somefile` را درون پوشه‌ی `data` قرار داده و آن را برای داوری ارسال نمایید. برای مثال، پوشه‌ی ارسالی شما برای داوری کوئرا به شکل زیر می‌تواند باشد:

```
.
├── data
│   └── somefile
└── docker-compose.yml
```

اگر محتوای پوشه `data` تان خالی است، می‌توانید این پوشه را ارسال نکنید.

توجه: سیستم داوری کوئرا به‌صورت خودکار فایل `docker-compose.yml` را با کامند `up` اجرا می‌کند. شما نیازی به کد یا اسکریپتی برای اجرای این کار ندارید.

نحوه ارسال پاسخ

شما فقط می‌توانید مسیرهای های‌لایت شده را تغییر دهید:

```
.
├── data
│   └── ...
├── docker-compose.yml
└── requirements.txt
```

در نهایت این پوشه را *zip* کرده و ارسال کنید. توجه کنید که پس از *extract* کردن فایل *zip* شما، باید فایل `docker-compose.yml` را ببینیم.