



طراحان: شهریار سلطان‌پور، مهدی فلاحي، رضا شکرچیان، سعید زنگنه، گلناز ادیب

مدرس: احسان خامس‌پناه

مهلت تحویل: سه‌شنبه ۶ خرداد ۱۳۹۹، ساعت ۲۳:۵۵



منابع زیرساختی که در این پروژه در اختیار شما قرار می‌گیرد، توسط ابر کافه‌بازار (زیرک) تامین شده است.

ما نیز از این طریق از حمایت‌هایشان تشکر و قدردانی می‌کنیم.

لقمه!

مقدمه

هدف از این پروژه، آشنایی با داکر، CI/CD و مدیریت کانتینرها به وسیله‌ی ابزار Kubernetes است. Continuous Integration یا یکپارچه‌سازی پیوسته فرایندی است که در آن به طور مداوم تغییرات تایید شده‌ی برنامه‌نویسان در یک خط اصلی ادغام شود. این کار به وسیله‌ی یک روند اتوماتیک انجام شده و کد جدید پس از بیلد شدن و اجرای تست‌ها به نسخه‌ی قابل اعتماد تری تبدیل می‌شود و اگر در این روند خطایی داشته باشد، تیم توسعه دهنده به سرعت از آن باخبر می‌شود. Continuous Delivery یا تحویل پیوسته فرایندی است که در آن پس از اطمینان از صحت کد جدید، طی فرایندی اتوماتیک یک نسخه‌ی جدید قابل انتشار باشد. در واقع نسخه‌ی آماده‌سازی شده در آن، در یک محیط تستی کوچک‌تر نسبت به محیط عملیاتی اصلی منتشر می‌شود تا عملکرد تغییرات جدید در محیطی واقعی مورد ارزیابی قرار بگیرد اما بسته به سیاست‌های گروه، روال انتشار آن در محیط عملیاتی اتومات نیست. Continuous Deployment یا استقرار پیوسته، علاوه بر مراحل فوق، نسخه‌ی جدید منتشر شده از برنامه را در محیط عملیاتی اصلی به طور اتوماتیک منتشر می‌کند. در این روش نیاز به سیستم‌های نظارتی بسیار محسوس است تا در صورت بروز مشکل تیم از آن مطلع شود. Kubernetes نیز ابزاری برای مدیریت سیستم‌های کانتینری در بستر ابری است که مدیریت، آپدیت‌دادن، scaling و ... را آسان و سریع می‌کند.

دقت کنید که همان‌طور که در کلاس درس گفته شده، مفاهیم CI/CD مستقل از داکر و کوبرنتیس تعریف شده‌اند. اما کاربردهای این دو ابزار پیاده‌سازی مفاهیم CI/CD را بسیار راحت و مقیاس‌پذیر کرده‌اند.

بخش اول: داکری کردن پروژهی React

در محیط‌های عملیاتی در برنامه‌های تحت REST، یک سرور وظیفه‌ی پاسخگویی به درخواست‌ها را دارد و یک سرور فایل‌های استاتیک (که پروژهی React شما نیز جزو این دسته قرار می‌گیرد) را ارائه می‌کند. شما باید یک داکرفایل بنویسید که ابتدا از پروژهی React یک بیلد برای محیط عملیاتی بگیرد و در یک سرور (مانند Nginx) فایل‌ها را ارائه دهد. این کار، مسأله‌ای مرسوم است و اگر در انجام آن به مشکل خوردید، سعی کنید سرچ کنید.

برای آشنایی با داکر و نوشتن داکرفایل [این لینک](#) را مطالعه کنید و برای Serve کردن پروژهی React توسط یک وب‌سرور (Nginx) می‌توانید [این لینک](#) را مطالعه کنید. برای این قسمت شما باید یک داکرفایل را که در پروژهی React شما اضافه شده تحویل دهید.

بخش دوم: داکری کردن پروژهی بک‌اند

شما در این بخش باید با نوشتن یک داکرفایل، یک ایمیج داکر برای سیستم بیلد شده‌ی خود ارائه بدید و در انتها با ساخته شدن یک کانتینر از ایمیج داکرفایل شما، سرور بک‌اند شما آماده به پاسخگویی باشد. دستوراتی که در داکرفایل تون می‌نویسید تقریباً همان دستوراتی است که روی سیستم در زمان اجرا و بیلد کردن پروژه استفاده می‌کنید. اگر در نوشتن داکرفایل به مشکلی خوردید حتماً به مستندات داکر مراجعه کنید و سعی کنید تا حد ممکن از راه‌حل‌هایی که برای مشکلاتتون در اینترنت وجود دارد استفاده کنید. برای این قسمت شما باید یک داکرفایل را که در پروژهی بک‌اند شما اضافه شده تحویل دهید.

بخش سوم: راه‌اندازی CI

در این بخش شما باید در محیط مخزن خود روال CI که در بخش مقدمه توضیح داده شد را پیاده کنید. با توجه به این که شما از تکنولوژی داکر استفاده می‌کنید، تنها کافی است که پس از هر تغییر در کد روی برنچ مستر، از روی داکرفایل پروژه، یک داکر ایمیج جدید بسازد. سپس آن را در یک رجیستری پوش کند تا در بخش‌های بعدی از آن استفاده کنید. برای اینکار می‌توانید از رجیستری‌هایی مانند [Dockerhub](#) یا [Quay.io](#) استفاده کنید. برای این قسمت شما باید فایل `gitlab-ci.yml` را که در پروژه‌های شما اضافه شده تحویل دهید. (پیشنهاد ما استفاده از ابزار گیت‌لب است اما اگر از ابزارهای دیگری برای این قسمت استفاده کنید مشکلی ندارد).

بخش چهارم: راه‌اندازی پروژه در محیط Kubernetes

برای این که بدانید کوبرنتیس چیست و چرا از آن استفاده می‌کنیم می‌توانید [این لینک](#) را مطالعه کنید و این [ویدیو](#) را ببینید. ابتدا در مورد مفاهیم Namespace، Pod، Service و Deployment در مورد Kubernetes بخوانید. (می‌توانید از [این لینک](#) استفاده کنید) برای شروع کار با کلاستران باید کلاينت کوبرنتیس (kubectl) را با کانفیگی که در اختیارتان قرار داده می‌شود راه‌اندازی کنید. (برای دریافت فایل کانفیگ گروه خود به shahryar.soltanpour@gmail.com ایمیل بزنید) بعد از تنظیم کردن کلاينت، به namespace خودتان برای ایجاد و تغییر ریسورس‌ها دسترسی خواهید داشت. سپس باید سه Deployment بسازید. یکی برای بک‌اند (دو Pod) و یکی برای فرانت‌اند (یک Pod) و یکی برای دیتابیس (یک Pod). برای اینکه پادهای بک‌اند و فرانت‌اند شما بالا بیایند باید کلاستر کوبرنتیس بتواند ایمیج شما را از رجیستری‌ای که ایمیج پروژه را در بخش ۳ در آن پوش کردید دریافت کند.

دقت کنید که با هر بار ریستارت شدن پاد دیتابیس، داده‌های ذخیره شده در آن نباید از بین بروند. (برای این کار مفاهیم persistent volume و persistent volume claim را مطالعه کنید. persistent volume ها از قبل ساخته شده‌اند و شما فقط باید persistent volume claim بسازید و به دیپلویمنت دیتابیس متصل کنید.)

در این قسمت شما هربار که می‌خواهید نسخه‌ی جدید کد را روی محیط Kubernetes قرار دهید، باید نسخه‌ی ایمج مربوطه را به صورت دستی در Deployment تغییر دهید.

دقت کنید که بخش‌های داخلی کوبرنتیس می‌توانند در شبکه‌ی داخلی آن همدیگر را ببینند اما برای دسترسی به آن‌ها از اینترنت خارجی، باید در kubernetes یک Service از نوع NodePort ایجاد کنید که برای هر Pod روی پورتهای که تعیین می‌کنید، دسترسی خارجی ایجاد شود.

در این قسمت شما حتماً باید در هنگام ساختن یک دیپلویمنت برای استفاده از memory و cpu محدودیت تعیین کنید، در غیر اینصورت پادهای شما ساخته نخواهند شد.

برای این قسمت شما باید فایل خروجی با فرمت yaml تمام ریسورس‌هایی که در kubernetes ساخته‌اید را در پروژه‌های بک‌اند و فرانت‌اند قرار دهید.

بخش پنجم: راه‌اندازی CD (امتیازی)

در قسمت پنجم، شما باید برای آپدیت پروژه در محیط Kubernetes، ایمج دیپلویمنت را به صورت دستی آپدیت می‌کردید. در این قسمت باید به CI پروژه که در بخش چهارم آن را پیاده کردید، یک stage جدید اضافه کنید تا بعد از هربار پوش کردن روی برنج مستر و موفقیت‌آمیز بودن Build پروژه، آخرین نسخه‌ی کد به طور اتوماتیک روی محیط پروداکشن قرار گیرد. (به دلیل مشکلاتی که رانرهای گیت‌لب در ارتباط با سرورهای داخل ایران دارند، ممکن است در این بخش دچار مشکل شوید و وقت زیادی از شما بگیرد)

نکات پایانی

- پروژه‌های زیرساختی آمیخته با مشکلات فراوان و پیش‌بینی نشده است. در صورتی که در هر یک از قسمت‌های پروژه بیش از حد گیر کردید و به بن‌بست رسیدید از دستیاران آموزشی کمک بگیرید.
- کافی است که یکی از اعضای گروه در دو خط آدرس مخزن‌های پروژه (back-end, front-end) را در گیت‌لب و در خط‌های بعدی Hash مربوط به آخرین کامیت‌های این مخزن‌ها را در سایت درس آپلود کند. در هنگام تحویل، پروژه روی این کامیت‌ها مورد ارزیابی قرار می‌گیرد.
- این تمرین یادگیری شماست. لطفاً تمرین را خودتان انجام دهید. در صورت مشاهده‌ی مشابهت بین کدهای دو گروه، ۵۰٪ نمره‌ی کل پروژه‌ها از گروه متقلب و تقلب‌دهنده کسر خواهد شد و در صورت تکرار، این روند ادامه خواهد داشت.
- سوالات خود را تا حد ممکن در فروم درس مطرح کنید تا سایر دانشجویان نیز از پاسخ آن‌ها بهره‌مند شوند. در صورتی که قصد مطرح کردن سوال خاص‌تری داشتید، از طریق ایمیل با طراحان این فاز پروژه ارتباط برقرار کنید. توجه داشته باشید که دیگر شبکه‌های اجتماعی مانند تلگرام راه ارتباطی رسمی با دستیاران آموزشی نیست و دستیاران آموزشی موظف به پاسخگویی در محیط‌های غیررسمی نیستند.
- ایمیل طراحان پروژه:

shahryar.soltanpour@gmail.com ●

mohammadf00900@gmail.com ●

rz.shekarchian@gmail.com ●

zangenehsaeed412@gmail.com ●

golnaz.a108@gmail.com ●