به نام خدا

رایانش ابری

تمرین دوم

داکر و مقدمات کوبرنتیز

طراحی تمرین:

آقایان توکلی، رجب‌پور و بوکانی

استاد درس:

آقای دکتر جوادی

مهلت نهایی ارسال پاسخ:

مقدمه

هدف از این تمرین، کار با داکر[1] و کوبرنتیز[2] است. بنابراین در این تمرین یک پروژه بسیار ساده را با استفاده از داکر کانتینرایز[3] می‌کنید و سپس بر روی سرور کوبرنتیز دیپلوی می‌کنید. برای پیش‌نیاز لازم است داکر و مینی‌کیوب[4] را بر روی سیستم خود نصب کرده باشید. برای راهنمایی می‌توانید از لینک‌های زیر استفاده کنید. دقت داشته باشید که برای اتصال به dockerhub و کار با Minikube نیاز به تنظیم شکن یا استفاده از فیلترشکن دارید.

https://shecan.ir/

https://docs.docker.com/get-docker/

https://minikube.sigs.k8s.io/docs/start/

گام اول

در ابتدا باید در داکرهاب[5] یک اکانت بسازید.

در این گام شما بایستی فعالیت‌های زیر را به ترتیب انجام بدید:

1. یک ایمیج داکر[6] بر پایه لینوکس (ترجیحا alpine ولی برای راحتی کار ubuntu هم مشکلی ندارد) بسازید که در آن امکان استفاده از دستور curl وجود داشته باشد.
2. ایمیج ساخته شده را بر روی داکرهاب آپلود نمایید.
3. برای تست کردن ایمیج ساخته شده، ایمیج خود را از داکرهاب دانلود کنید و یک کانتینر از آن بالا بیاورید.
4. یک درخواست curl به google.com ارسال کنید.

موارد زیر را در فایل گزارش نمایش دهید:

- ارسال ایمیج ساخته شده بر روی داکرهاب و نتیجه آن
- نمایش لیست ایمیج های موجود بر روی سیستم خود
- دریافت ایمیج ساخته شده از داکرهاب
- ساختن کانتینر از ایمیج دریافت شده از داکرهاب
- اجرا دستور curl و نتیجه آن

---

[1] Docker
[2] Kubernetes
[3] Containerize
[4] Minikube
[5] https://hub.docker.com/
[6] Docker Image

گام دوم

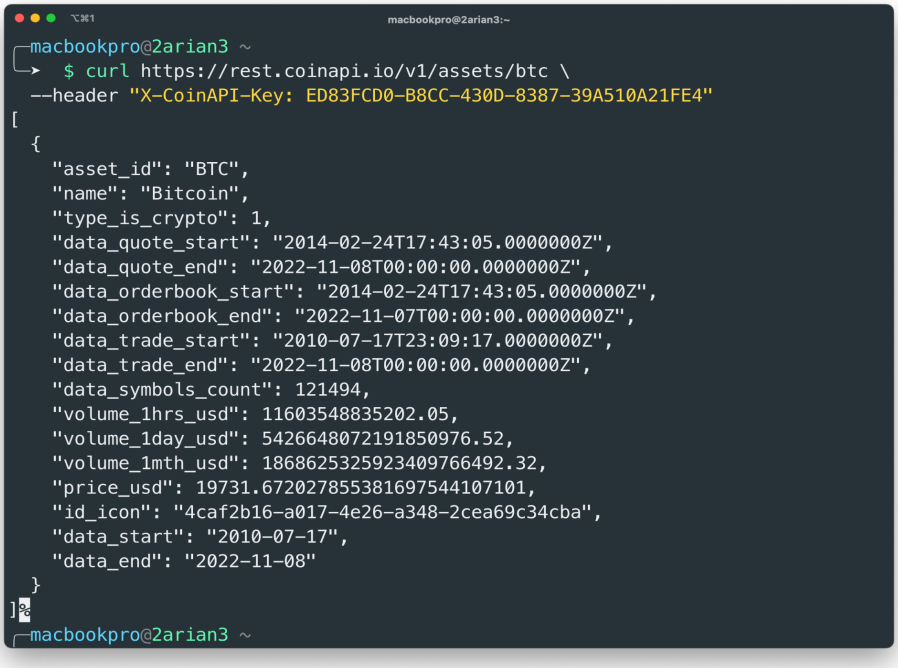
در این گام قرار است که سروری را برای دریافت اطلاعات مربوط به رمز ارزها توسعه دهیم و به صورت مشخص، قیمت کنونی آنها را نگهداری کنیم و برگردانیم. محدودیتی در API استفاده شده وجود ندارد و بسته به سلیقه خود می‌توانید از هر API موجود برای دریافت اطلاعات استفاده کنید. در قسمت زیر لیست API های پیشنهادی آورده شده‌اند:

- Coin.api
- Coingecko.com
- Kucoin.com

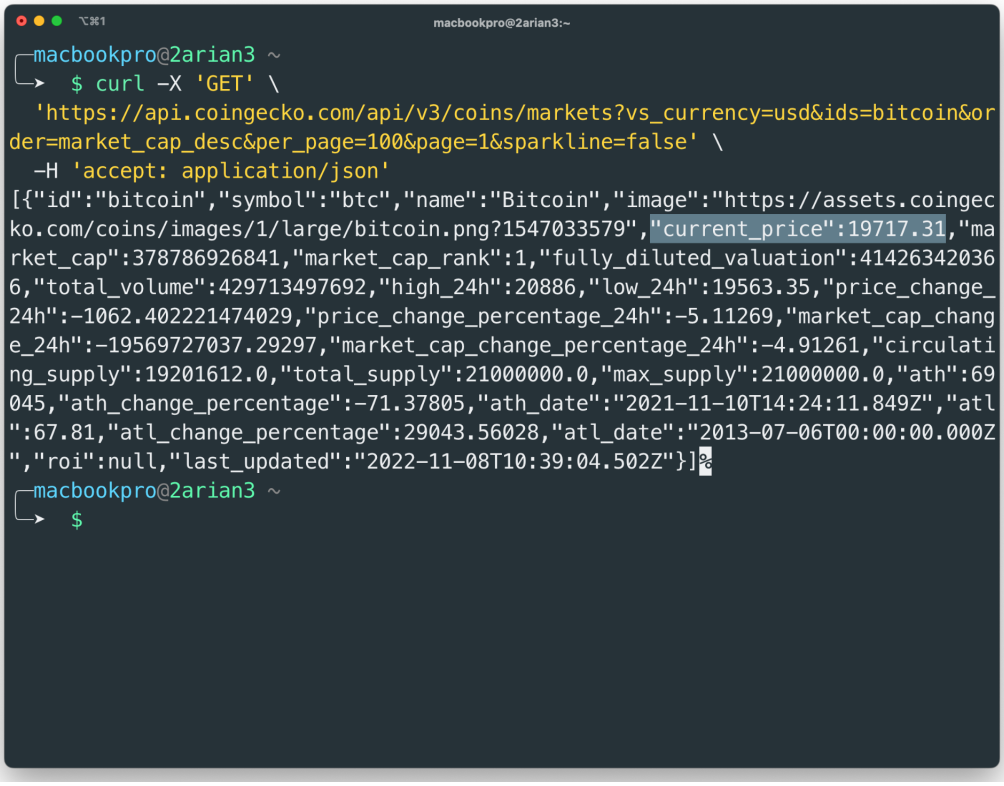
توجه :چنانچه از مورد اول استفاده می‌کنید، نیاز به ساختن یک API Key و اضافه کردن آن به عنوان هدر به درخواست HTTP خود دارید.

توجه :با هر زبان برنامه‌نویسی دلخواه می‌توان سرور را توسعه داد.

در شکل‌های زیر، نمونه‌هایی از این درخواست‌ها آورده شده‌اند.



شکل ۱ نمونه‌ای از درخواست API مربوط به coin.api برای دریافت قیمت BTC



شکل ۲ نمونه‌ای از درخواست API مربوط به coingecko.com برای دریافت قیمت BTC

در شکل زیر هم یک نمونه‌ی ساده از خروجی‌ای که سرور توسعه داده شده به ازای هر درخواست باید داشته باشد، آورده شده است.

شکل ۳ ساختار خروجی هر درخواست به سرور توسعه داده‌شده

همچنین برای جلوگیری از ارسال درخواست های اضافه در برنامه خود باید از کش ردیس[7] استفاده کنید. برای استفاده از ردیس، ایمیج آن را دانلود کنید و یک کانتینر از آن بالا بیاورید. در کلید اسم ارز و در مقدار قیمت آن ارز را نگه دارید و مقادیر باید به صورت پیش‌فرض به مدت پنج دقیقه اعتبار داشته باشند.

[7] Redis

حال باید برای برنامه خود یک Dockerfile بنویسید و آن را build کنید و ایمیج خود را بسازید و بر روی داکرهاب قرار دهید.

توجه: در این مرحله شما باید دو کانتینر داشته باشید. یکی مربوط به کش ردیس و دیگری مربوط به برنامه خود.

راهنمایی: برای برقراری ارتباط با کش ردیس از داخل برنامه خود، یا یک شبکه[8] بین این دو کانتینر تعریف کنید تا کانتینرها بتوانند همدیگر را ببینند و با هم در ارتباط باشند.

می‌دانیم با پایین آمدن کانتینر ردیس، اطلاعات از دست خواهند رفت. برای جلوگیری از این اتفاق، باید اطلاعات کش ردیس را persist کنیم تا با پایین آمدن، اطلاعات از بین نروند و پس از بالا آمدن مجدد اطلاعات قبلی بازیابی شوند. برای این منظور باید از قابلیت Volume استفاده کنید.

نکته مهم: پروژه باید برای فیلد های زیر کانفیگ پذیر باشد (نباید در کد به صورت hard code تعریف شوند)

- شماره پورتی که بر روی آن سرور خود را بالا می‌آورید
- مدت زمان اعتبار کلیدها در کش ردیس
- نام رمز ارزی[9] که قیمت آن را می‌خواهیم
- API Key (در صورتی که از API مورد اول استفاده می‌کنید)

موارد زیر را در فایل گزارش نمایش دهید:

- دریافت ایمیج ردیس و ساختن کانتینر از آن
- ساختن شبکه برای برقراری ارتباط بین دو کانتینر
- ساختن Volume جهت persist کردن اطلاعات کش ردیس
- ساختن ایمیج سرور نوشته شده با استفاده از داکرفایل
- ارسال ایمیج ساخته شده بر روی داکرهاب و نمایش نتیجه آن
- نمایش اطلاعات ایمیج سرور خود با استفاده از دستور docker inspect
- نمایش کانتینرهای موجود در سیستم خود با استفاده از دستور docker ps
- نمایش میزان منابع استفاده شده توسط کانتینر های موجود با استفاده از دستور docker stats

[8] network
[9] Cryptocurrency

گام سوم

در این بخش هدف کار با یک ابزاری است که با آن می‌توانید یک کلاستر کوبرنتیز را به صورت ساده و سریع بر روی سیستم خود بالا بیاورید. این ابزار Minikube[10] نام دارد. شما باید برای سرور خود یک سری فایل‌های مورد نیاز جهت دیپلوی کردن آن بر روی کلاستر کوبرنتیز بنویسید.

- فایل ConfigMap: در این فایل یک سری تنظیمات برنامه مانند آدرسی که برنامه باید به آن درخواست ارسال کند، پورت سرور، نام رمز ارز و مدت زمان نگه‌داری کش و API key مشخص می‌شود (برای خوانایی بهتر این فایل، می‌توانید یک فایل کانفیگ جداگانه بنویسید و آدرس آن را به فایل ConfigMap بدهید).
- فایل Deployment: جهت مدیریت کردن پادها می‌باشد. در این فایل تعداد replica برنامه را برابر با ۲ قرار دهید.
- فایل Service: این فایل جهت ایجاد دسترسی به سرور می‌باشد.

حال که تمام فایل‌ها را ساختید آن‌ها را به همین ترتیب ساختشان، با استفاده از دستور kubectl apply بر روی Minikube اعمال کنید.

حال این ۳ فایل را برای دیتابیس ردیس نیز به صورت جداگانه بنویسید با این تفاوت که تعداد replica را در فایل Deployment برابر با ۱ قرار دهید. برای کش ردیس علاوه بر فایل‌های بالا نیاز به دو فایل دیگر جهت نگه‌داری اطلاعات دارید.

- فایل Persistent Volume: با این فایل حافظه را در کلاستر مشخص می‌کنید.
- فایل Persistent Volume Claim: با این فایل درخواستی برای اختصاص حافظه‌ای -که با فایل قبل در کلاستر ایجاد کردید- برای یک کانتینر ایجاد می‌کنید.

دوباره همانند قبل به ترتیب فایل‌ها را با دستور kubectl apply بر روی کلاستر اعمال کنید.

موارد زیر را در فایل گزارش نمایش دهید:

- با استفاده از دستور kubectl get صحت ایجاد منابع (کانفیگ‌مپ، دیپلویمنت، سرویس و پادها، PVC، PV) را بر روی کلاستر Minikube نشان دهید.
- همچنین آدرس IP پادها را با استفاده از endpoint نشان دهید.

[10] https://minikube.sigs.k8s.io/docs/start/

گام چهارم

در این بخش می‌خواهیم صحت سیستم را تست کنیم. ابتدا ایمیجی که در گام اول (که با آن قابلیت ارسال درخواست[11] داشتید) ساخته بودید را بر روی کلاستر Minikube با استفاده از دستور kubectl run اجرا کنید. سپس به سرویسی که برای سرور خود ساختید درخواست بزنید.

موارد زیر را در فایل گزارش نمایش دهید:

- با استفاده از دستور kubectl get صحت ایجاد منابع را بر روی کلاستر Minikube نشان دهید.
- از آنجایی که تعداد replica ها را در فایل Deployment سرور خود بیش از ۱ قرار داده‌اید، با ارسال درخواست‌های مکرر به سرویس سرورتان، باید توزیع بار میان پادها را مشاهده کنید. این توزیع بار را نشان دهید (که درخواست تنها به یک پاد ارسال نمی‌شود).

[11] Request

نکات مربوط به تحویل تمرین

- تمرین شما تحویل اسکایپی خواهد داشت. بنابراین از استفاده از کدهای یکدیگر یا کدهای موجود در وب که قادر به توضیح داده عملکرد آنها نیستید، بپرهیزید!
- در تحویل اسکایپی از شما خواسته می‌شود تا با استفاده از فایل های دیپلویمنت نوشته شده، پروژه خود را از صفر بر روی کلاستر کوبرنتیز ببرید و در تعداد پادهای آن تغییر ایجاد کنید. همچنین باید بلد باشید پس از تغییر فایل کانفیگ، آن را بر روی پادها اعمال کنید.
- ابهامات خود را در سایت درس مطرح کنید و ما در سریع‌ترین زمان ممکن به آنها پاسخ خواهیم داد.

آنچه که باید ارسال کنید

یک فایل زیپ با نام sid_HW2.zip که شامل موارد زیر است (هر مورد را در فولدر جداگانه قرار دهید)

- برای گام اول لازم است یکی از موارد زیر آپلود شود
  1. Dockerfile نوشته شده برای ساخت ایمیج مورد نظر
  2. تمامی دستوراتی که برای ساخت ایمیج مورد نظر از طریق docker commit انجام داده‌اید.
- برای گام دوم تمامی فایل های پروژه به همراه Dockerfile آن و فایل config آن
- برای گام سوم فایل های دیپلویمنت
- گزارشی که حداقل باید شامل موارد مطرح شده در توضیحات تمرین (به همراه اسکرین‌شات) باشد

موفق باشید

تیم درس مبانی رایانش ابری