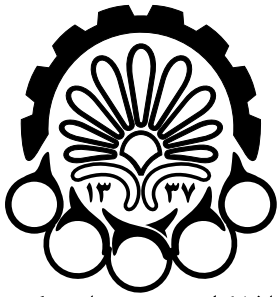
به نام خدا



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

رایانش ابری

پروژه پایانی
داکر و کوبرنتیز

طراحی پروژه:

آقای حسینی

استاد درس:

آقای دکتر جوادی

مهلت نهایی ارسال پاسخ:

۶ بهمن ماه ۱۴۰۰ ساعت ۲۳:۵۹

نکته مهم: دقت کنید که تمدید نخواهیم.

مقدمه

هدف از این تمرین کار با داکر و کوبرنتیز به صورت اصولی است. لذا قصد داریم یک پروژه نسبتا ساده را با استفاده از داکر در ابعاد مناسب containerize کرده و بر روی کوبرنتیز دیپلوی کنیم. برای انجام این تمرین لازم است docker و minikube را بر روی سیستم خود نصب کرده باشد. به این منظور می‌توانید از لینک‌های زیر کمک بگیرید:

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

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

گام اول

در این گام قصد داریم یک پروژه URL shortener توسعه دهیم. چند مثال از این پروژه را می‌توانید از طریق لینک‌های زیر مشاهده کنید:

https://www.shorturl.at/

https://cutt.ly/

در این پروژه هر کاربر می‌تواند لینک مورد نظر خود را وارد کند و سرور یک لینک کوچک و خلاصه شده تحویل دهد تا با استفاده از آن بتوان به آدرس لینک اولیه دسترسی داشت. به این منظور شما باید یک سرور به زبان دلخواه خود توسعه دهید. این سرور با یک دیتابیس (پیشنهاد ما MongoDB است) در ارتباط است که آدرس‌های کوتاه شده‌ی **یکتا** را در آن ذخیره می‌کند. از آنجایی که این خدمت ارائه شده رایگان است، آدرس‌های ذخیره شده دارای تاریخ انقضا هستند و به صورت دائمی قابل دسترسی نیستند. سرور شما دارای دو endpoint است که یکی از آن‌ها وظیفه ایجاد آدرس کوتاه شده و دیگری وظیفه انتقال به آدرس کوتاه شده را دارد.

برای مثال با ارسال یک درخواست HTTP و با متد Post به اندپوینت زیر، سرور آدرس کوتاه شده را برای ما ارسال می‌کند.

```
~
└─ curl --location --request POST 'https://www.shorturl.at/shortener.php' \
--form 'u="www.example-webiste.com%2Fa-really-really-long%2Faddress"'
```

فرض کنید جواب سرور در این حالت آدرس کوتاه شده «shorturl.at/nsBL5» است.

و با ارسال در خواست به اندپوینت دوم که همان آدرس کوتاه شده‌است، آدرس اولیه طولانی برای ما بارگذاری می‌شود.

```
~
└─ curl --location --request GET 'shorturl.at/nsBL5'
```

دقت داشته باشید این آدرس کوتاه شده برای مدت زمان محدودی قابل دسترسی است و پس از آن منقضی می‌شود.

نکته مهم: پروژه شما باید مانند تمرین دوم کانفیگ پذیر باشد. فیلدهای زیر از طریق فایل کانفیگ مقدار دهی می‌شوند:

- شماره portای که سرور بر روی آن اجرا می‌شود
- مدت زمان انقضای آدرس URL کوتاه شده
- آدرس سرور دیتابیس ساخته شده
- اسم و رمز عبور دیتابیس مورد استفاده

پروژه شما می‌تواند به هر زبانی توسعه داده شود.

گام دوم

پس از اتمام پیاده‌سازی، برای پروژه خود یک Dockerfile بنویسید که با استفاده از آن، بتوان پروژه را containerize کرد. در نهایت با build کردن Dockerfile ایمیج پروژه خود را تولید کرده و بر روی داکرهاب قرار دهید.

نکته مهم: شما باید از تکنیک **multistage build** کمک بگیرید و در دو مرحله ایمیج خود را تولید کنید. وظیفه مرحله اول تنها build کردن پروژه شما و ساخت فایل قابل اجرا است تا نهایتا در مرحله دوم این فایل در یک کانتینر **alpine** اجرا شود.

موارد زیر را در فایل گزارش نمایش دهید:

۱) build کردن ایمیج با استفاده از Dockerfile ساخته شده

۲) ارسال ایمیج ساخته شده بر روی داکرهاب و نتیجه آن

۳) در صورتی که پروژه خود را با استفاده از ایمیج ساخته شده بر روی سیستم شخصی خود تست کردید، تصاویر مربوطه را قرار دهید (این مرحله اجباری نیست ولی توصیه می‌شود)

۴) محتویات Dockerfile

گام سوم

حال زمان این است که با نوشتن فایل‌های دیپلویمنت کوبرنتیز، پروژه خود را بر روی minikube بالا بیاوریم.

اولین کامپوننت مورد نیاز یک ConfigMap برای پروژه است تا بتوان port سرور، زمان انقضای آدرس‌های ایجاد شده و آدرس سرور دیتابیس از آن خوانده شود.

کامپوننت بعدی یک Secret است که وظیفه ذخیره سازی اسم و رمز عبور دیتابیس را بر عهده دارد. از آن جایی که این اطلاعات، مخصوصا رمز عبور، جزو اطلاعات حساس هستند باید آن‌ها را در Secret ذخیره نمود.

به منظور پایدار ماندن اطلاعات دیتابیس در صورت بروز مشکل برای پادهای مربوطه، لازم است تا برای آن Persistent Volume تعریف کنیم. در نتیجه گام بعدی ایجاد Persistent Volume و در ادامه ساخت Persistent Volume Claim برای استفاده از آن است.

سپس باید یک توصیف deployment بنویسید که وظیفه آماده سازی و نگهداری از دیتابیس را بر عهده دارد (نحوه ایجاد دیتابیس به انتخاب شماست. تنها نکته مهم برخورداری از رمز عبور تعریف شده در توصیف Secret است). فراموش نکنید که

Persistent Volume Claim ساخته شده در مرحله قبل را بر این دیپلویمنت سوار کنید (به نظر شما تعداد پادهای مناسب این توصیف چند عدد است؟).

برای دسترسی به این دیتابیس به یک Service نیاز است که با استفاده از آن می‌توانیم ارتباط پروژه و دیتابیس را قرار کنیم.

حال می‌توان یک Deployment نوشت که وظیفه آماده سازی و نگهداری پادها را بر عهده دارد. تعداد replica را برابر با ۲ تعیین کنید (دقت داشته باشید که در این توصیف شما باید Secret و ConfigMap ساخته شده را در اختیار پروژه قرار دهید تا مقادیر لازم از طریق آن‌ها پر شود).

آخرین مورد یک Service است که با استفاده از آن می‌توانیم به پروژه و در واقع سروری که توسعه داده‌ایم دسترسی داشته باشیم.

پس از ساخت فایل‌های گفته شده، آن‌ها را به همان ترتیب و با استفاده از دستور kubectl apply بر روی کلاستر minikube ایجاد کنید.

موارد زیر را در فایل گزارش نمایش دهید:

۱) با استفاده از دستور kubectl get صحت ایجاد منابع بر روی کلاستر را نمایش دهید

۲) آدرس IP پادها و نحوه برقراری ارتباط میان آن‌ها و سرویس ساخته شده

۳) برای دیپلویمنت مربوط به دیتابیس چه تعداد پاد ایجاد کردید؟ دلیل کار خود را توضیح دهید

موارد امتیازی

در ادامه مواردی مطرح می‌شوند که بر خلاف گام‌های تعریف شده، در کلاس به صورت کامل به آن‌ها پرداخته نشد و تنها اشاره مختصری صورت گرفت. لذا لازم است تا برای انجام آن‌ها کمی در اینترنت جستجو کنید (توجه داشته باشید که مورد اول بسیار ساده‌است و توصیه می‌شود آن را انجام دهید).

ساختن یک کامپوننت HPA در کلاستر کوبرنتیز به منظور انجام عملیات auto scaling

موارد زیر را در فایل گزارش نمایش دهید:

۱) پارامترهای موجود جهت مقیاس کردن خودکار را بیان کنید

۲) شما کدامیک از این پارامترها را برای ایجاد HPA استفاده کردید؟ دلیل خود را شرح دهید

۳) دستور و یا توصیف مورد استفاده برای ساخت HPA

اجرای دیتابیس خود با استفاده از توصیف stateful set و جایگزین کردن آن با deployment (دقت داشته باشید این کار نباید با کمک گرفتن از helm chart آماده انجام شود)

موارد زیر را در فایل گزارش نمایش دهید:

۱) دلیل استفاده از stateful set بجای deployment

۲) توصیف مورد استفاده برای ساخت stateful set

پیاده‌سازی helm chart جهت خودکار سازی ایجاد منابع و توصیف‌های تعریف شده، بر روی کلاستر کوبرنتیز

موارد زیر را در فایل گزارش نمایش دهید:

۱) توضیح مختصر ساختار helm chart

پیاده‌سازی docker compose جهت خودکار سازی ایجاد منابع و وابستگی‌های مورد نیاز پروژه و نهایتا build و اجرای آن

موارد زیر را در فایل گزارش نمایش دهید:

۱) محتویات و توضیح مختصر docker compose پیاده‌سازی شده

تست پروژه

در نهایت باید با استفاده از یکی از موارد زیر، پروژه را تست کرده و خروجی را گزارش خود قرار دهید:

- با استفاده از port forwarding سرویس ایجاد شده برای پروژه
- با استفاده از یک image با قابلیت curl (مانند آنچه در تمرین دوم انجام شد)

نکات مربوط تحویل به پروژه

- پروژه شما تحویل اسکایپی خواهد داشت بنابراین از استفاده از کدهای یکدیگر یا کدهای موجود در وب که قادر به توضیح دادن عملکرد انها نیستید، بپرهیزید.
- در تحویل اسکایپی از شما سوال‌هایی در رابطه با فایل‌های دیپلویمنت نوشته شده، نحوه ارتباط آن‌ها با یکدیگر و ایجاد تغییر روی تعداد و تنظیمات پاده‌ها می‌شود. همچنین باید بلد باشید پس از تغییر فایل کانفیگ آن را بر روی پادها اعمال کنید.
- ابهامات خود را در سایت و یا گروه تلگرامی درس مطرح کنید و ما در سریعترین زمان ممکن به انها پاسخ خواهیم داد.

آنچه که باید ارسال کنید

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

- تمامی فایل‌های پروژه
- گزارش که حداقل باید شامل موارد مطرح شده در توضیحات پروژه باشد

موفق باشید

تیم درس مبانی رایانش ابری