

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:

۱۱ بهمن ماه ۱۴۰۲ ساعت ۲۳:۵۹

مقدمه

هدف از این تمرین کار با داکر و کوبرنتیز به صورت اصولی است. لذا قصد داریم یک پروژه نسبتاً 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 http monitoring system توسعه دهیم. وظیفه این سیستم بررسی وضعیت سلامت وب سرورها است. هر کاربر می‌تواند آدرس وبسایت مد نظر خود را در این سیستم ثبت کند تا به صورت پیاپی و در بازه‌های زمانی مشخص، وضعیت سلامت آن بررسی شود.

به صورت کلی شما باید سروری شامل سه endpoint زیر توسعه دهید:

POST: /api/server

GET: /api/server?id=

GET: /api/server/all

برای ساده‌سازی احراز هویت برای کاربران تعریف نشده است. کاربر می‌تواند با ارسال درخواست POST به اندپوینت اول، وب‌سرور مورد نظر خود را اضافه کند. در این صورت سرور شما آدرس مشخص شده توسط کاربر را در دیتابیس ذخیره می‌کند و یک id به نشانه شماره سرور ذخیره شده به کاربر برمیگرداند:

id	address	success	failure	last_failure	created_at
1	216.239.38.120	20	2	1703852574	1703840374
2	aut.ac.ir	30	50	1702840374	1700840374

سرور شما باید با توجه به زمان مشخص شده در کانفیگ (برای مثال هر ۶۰ دقیقه) لیست وب‌سرورها را از دیتابیس گرفته، و وضعیت سلامت هر کدام را با ارسال یک درخواست GET به آدرس آن بررسی کند. در صورتی که پاسخ وب‌سرور در بازه‌ی موفقیت ۲۰۰ باشد، تعداد درخواست‌های موفق (success) یک واحد افزایش می‌یابد و در غیر این صورت، با آپدیت شدن last_failure، تعداد درخواست‌های ناموفق یک واحد افزایش پیدا می‌کند. (زمان به صورت timestamp ذخیره شده است)

دو اندپوینت بعد به منظور مشاهده نتایج ذخیره شده تعریف می‌شوند. کاربر می‌تواند با مقدار دهی id وب‌سرور تعریف شده وضعیت وب‌سرور مورد نظر را مشاهده کند و یا با ارسال درخواست به اندپوینت آخر وضعیت تمامی وب‌سرورهای تعریف شده را دریافت کند. دقت داشته باشید که پاسخ سرور شما باید خوانا باشد. (توصیه می‌شود پاسخ سرور با فرمت json باشد)

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

- شماره port ای که سرور بر روی آن اجرا می‌شود
 - بازه‌های زمانی بررسی سلامت (می‌تواند به صورت الگوی cron و یا زمان به دقیقه تعریف شود)
 - دو آدرس سرور دیتابیس ساخته شده برای خواندن و نوشتن به صورت مجزا (این مورد در گام دوم و تعریف Dockerfile شامل دو آدرس یکسان بوده و در زمان تعریف statefulset کاربرد دارد)
 - 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 نهایتاً در مرحله دوم این فایل در یک کانترینر **alpine** اجرا شود.

در ادامه باید یک docker compose کامل شامل volume و network، جهت خودکار سازی ایجاد منابع و وابستگی‌های مورد نیاز پروژه پیاده‌سازی کنید که نهایتاً build و اجرای آن را انجام می‌دهد.

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) صحت اجرا شدن داکر کامپوز و مراحل آن

گام سوم

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

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

سپس باید یک توصیف statefulset 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). دقت داشته باشید که در این بخش service‌های مجزایی برای عملیات read و write ساخته می‌شوند که استفاده صحیح از آن‌ها ضروری و دارای نمره است. در این مرحله باید مقداردهی معقولی برای resource‌های مورد استفاده دیتابیس خود داشته باشید و request و limit آن را تعریف کنید.

کامپوننت بعدی یک ConfigMap برای پروژه است تا بتوان port سرور، زمان بررسی وضعیت آدرس‌های ایجاد شده و آدرس‌های سرور دیتابیس از آن خوانده شود.

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). در این مرحله باید مقداردهی معقولی برای resource‌های مورد استفاده برنامه خود داشته باشید و request و limit آن را تعریف کنید.

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

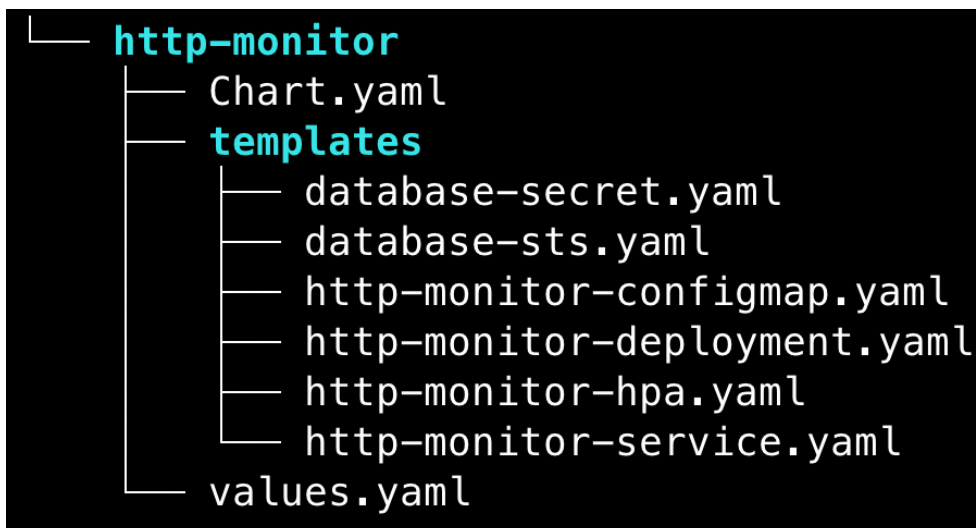
(۲) آدرس IP پادها و نحوه برقراری ارتباط میان آن‌ها و سرویس ساخته شده

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

(۴) نحوه استفاده از سرویس مستر و رپلیکاها

گام چهارم

در این گام قصد داریم تا با آماده‌سازی helm chart، مراحل استقرار بر روی کلاستر را اصولی‌تر و ساده‌تر کنیم. شما باید ساختار زیر را برای پروژه ایجاد کنید و موارد لازم را به صورت صحیح مقدار دهی کنید:



مقدار دهی صحیح values بسیار مهم است.

موارد زیر را در فایل گزارش نمایش دهید:

(۱) با استفاده از دستور helm install سرور خود را بر روی کلاستر استقرار دهید

(۲) صحت ایجاد منابع را نشان داده و سپس با دستور helm uninstall منابع خود را حذف کنید

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 ساده است و توصیه می‌شود آن را انجام دهید). در صورت انجام هر کدام از موارد زیر، در گزارش خود موارد زیر آن‌ها را توضیح دهید.

ساختن یک کامپوننت HPA در کلاستر کوبرنتیز به منظور انجام عملیات auto scaling. برای صحت عملکرد این مورد باید یک «metric server» روی کلاستر خود راه‌اندازی کنید. (۲۰ درصد)

(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

(۴) صحت مقیاس سرور شما در صورت افزایش بار روی آن

تعریف های probe, liveness, readiness و startup به شکل صحیح. از این سه به منظور مشخص کردن سلامت پادهای اپلیکیشن استفاده می‌شود. شما باید با استدلال درست هر کدام را به شکلی تعریف کنید که وظیفه هر کدام به درستی انجام پذیرد. (۳۰ درصد)

(۱) توضیح روش کار و تاثیر هر کدام بر اپلیکیشن

(۲) استدلال و منطق پشت هر یک از پیاده‌سازی‌های شما

استفاده از prometheus و تعریف metricهایی که حداقل شامل تعداد درخواست، تعداد درخواست‌های همراه با خطا، مدت زمان پاسخ به هر درخواست، تعداد خطاهای دیتابیس و مدت زمان پاسخ دیتابیس باشد. (۳۰ درصد)

(۱) عنوان و نوع metricهای تعریف شده

(۲) یک مورد کوئری به ازای هر کدام از metricهای جمع‌آوری شده

استفاده از grafana و تعریف dashboardهایی به منظور تصویرسازی و مشاهده بهتر metricهای تعریف شده در مرحله قبل. (۲۰ درصد)

(۱) کوئری‌های تعریف شده برای هر dashboard

(۲) تصویر تمامی dashboardها

تست پروژه

باید با استفاده از port forwarding سرویس ایجاد شده برای پروژه، آن را تست کرده و خروجی را گزارش خود قرار دهید.

نکات مربوط تحویل به پروژه

- پروژه شما تحویل اسکایی خواهد داشت بنابراین از استفاده از کدهای یکدیگر یا کدهای موجود در وب که قادر به توضیح دادن عملکرد آنها نیستید، بپرهیزید.
- در تحویل اسکایی از شما سوال‌هایی در رابطه با فایل‌های توصیف نوشته شده، نحوه ارتباط آن‌ها با یکدیگر و ایجاد تغییر روی تعداد و تنظیمات پادها می‌شود. همچنین باید بتوانید 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 آنها 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