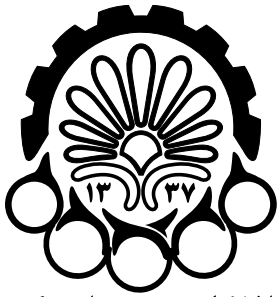
به نام خدا



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

رایانش ابری

پروژه پایانی
داکر و کوبرنتیز

طراحی پروژه:

آقایان احمدوند، فرنگی‌زاده و حسینی

استاد درس:

آقای دکتر جوادی

مهلت نهایی ارسال پاسخ:
۱۱ تیر ۱۴۰۱ ساعت ۷:۳۰ صبح

نکته مهم: دقت کنید که تمدید نخواهیم (به دلیل مهلت نهائی کردن نمرات).

مقدمه

هدف از این تمرین درسی، 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 private note توسعه دهیم. یک مثال از این پروژه را می‌توانید از طریق لینک زیر مشاهده کنید:

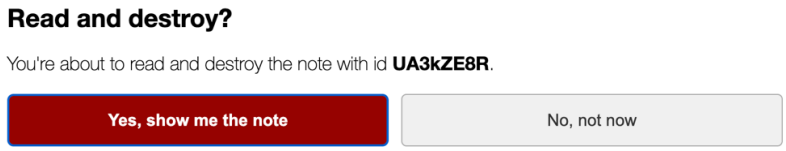
https://privnote.com/

در این پروژه هر کاربر می‌تواند متن (یادداشت) مورد نظر خود را وارد کند و سرور یک آدرس تحویل دهد تا با استفاده از آن بتوان به یادداشت دسترسی داشت. هر شخصی می‌تواند از طریق آن لینک به یادداشت دسترسی داشته باشد اما پس از آنکه اولین نفر یادداشت را باز کرد، یادداشت پاک شده و دیگر قابل دسترسی نخواهد بود. به این منظور شما باید یک سرور به زبان دلخواه خود توسعه دهید. این سرور با یک دیتابیس در ارتباط است که یادداشت‌ها و آدرس‌های **یکتایشان** 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.

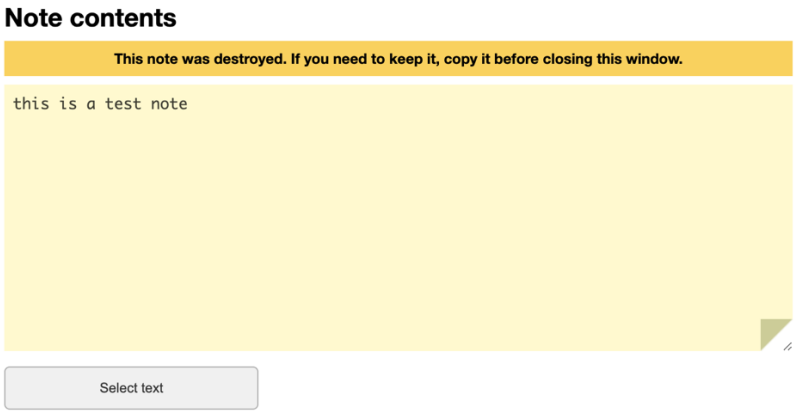
سرور شما دارای چهار endpoint است. برای سرور خود یک «front-end» مینیمال طراحی کنید. اندپوینت اول، صفحه اصلی را نمایش می‌دهد:

با ارسال یک درخواست HTTP و با متد Post به اندپوینت دوم، سرور آدرس یادداشت ساخته شده را برای ما ارسال می‌کند.

و با ارسال درخواست به اندپوینت سوم که همان آدرس است، پیام زیر به ما نمایش داده می‌شود:



در صورت تایید، یادداشت نمایش داده شده و حذف می‌گردد (توسط اندپوینت آخر):



دقت داشته باشید این آدرس 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 پورتی که سرور بر روی آن اجرا می‌شود.
- مدت زمان انقضای آدرس URL
- آدرس سرور دیتابیس ساخته شده
- اسم و رمز عبور دیتابیس مورد استفاده

نکته: 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. به عبارت دیگر اندازه ایمیج نهائی بایستی مینیمال باشد.

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 پورت سرور، زمان انقضای آدرس‌های ایجاد شده و آدرس سرور دیتابیس از آن خوانده شود.

کامپوننت بعدی یک Secret است که وظیفه ذخیره‌سازی اسم و رمز عبور دیتابیس را بر عهده دارد. از آن جایی که این اطلاعات، مخصوصا رمز عبور، جزء 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 آماده‌سازی 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 آماده‌سازی 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). 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.

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). توجه داشته باشید این مورد همراه با تغییراتی در کد پروژه شما است تا به درستی از master و slave ساخته شده استفاده شود.

۱) دلیل استفاده از stateful set بجای deployment

۲) توصیف مورد استفاده برای ساخت stateful set

۳) نحوه استفاده از سرویس مستر و رپلیکاها

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

۱) توضیح مختصر ساختار helm chart

۲) محتویات و توضیح مختصر پارامترهای تعریف شده در فایل values مربوط به چارت (تعریف درست پارامترها بسیار مهم است)

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

۱) محتویات و توضیح مختصر docker compose پیاده‌سازی شده

أزمون 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 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