مبانی رایانش ابری

تمرین دوم (فاز ۳)

kubernetes

طراحان تمرین:

علی بازشوشتری، ابوالفضل حسینی

استاد درس:

دکتر جوادی

مهلت نهایی ارسال پاسخ:

۱۰ اردیبهشت ساعت ۲۳:۵۹

مقدمه:

در فاز آخر از تمرین کوبرنتیز[1] قصد داریم یک کلاستر کوبرنتیز بالا آورده و با مولفه‌های مختلف کوبرنتیز مانند Deployment، Service و ConfigMap آشنا شویم.

بخش ۱

Deployment

برای دیپلوی[2] کردن یک برنامه بر روی کلاستر کوبرنتیز[3] شما می‌توانید مستقیما از دستوری مانند `kubectl run` استفاده کنید. این دستور مستقیما یک Pod بر روی کلاستر کوبرنتیز می‌سازد (اطلاعات بیشتر). اما اعمال کردن مستقیم یک برنامه بر روی کلاستر از اهداف اصلی کوبرنتیز مانند auto scaling پشتیبانی نمی‌کند. در نتیجه فقط در موارد خاص مثل زمانی که قرار است یک برنامه را تست کنیم یا زمانی که اجرای برنامه به صورت مداوم برای ما اهمیتی ندارد، از این شیوه برای بالا آوردن یک پاد[4] بر روی کلاستر کوبرنتیز استفاده می‌شود.

برای دیپلوی کردن یک برنامه بر روی کلاستر کوبرنتیز از یک فایل مخصوص Deployment با پسوند yaml. استفاده می‌کنیم. این فایل شرایط مطلوب[5] ما برای برنامه را در اختیار کوبرنتیز قرار می‌دهد و Kubernetes Plane Control تضمین می‌کند که وضعیت کلاستر همیشه به همین شکل باقی بماند؛ به عبارتی Desired State و Current State همیشه یکسان باشند. برای مثال اگر تعداد replica در فایل Deployment، سه تا تعریف شده باشد و یکی از پادها خاتمه یابد، Control Plane به صورت خودکار یک پاد جدید می‌سازد تا تعداد replicaها سه تا باقی بماند.

شما برای دیپلوی کردن برنامه جستجوی فیلم خود که در فاز 1 تمرین دوم (بخش داکر) نوشتید، لازم است که برای هر کدام از API Server خود، Elasticsearch و Redis یک فایل Deployment با پسوند yaml. بنویسید. می‌توانید نمونه‌ای از این فایل Deployment را در اینجا مشاهده نمایید.

در هرکدام از فایل‌های دیپلویمنت مرتبط با سرور خود و الستیک، تعداد replica را 3 و در فایل دیپلویمنت ردیس تعداد replica را برابر 1 قرار دهید.

---

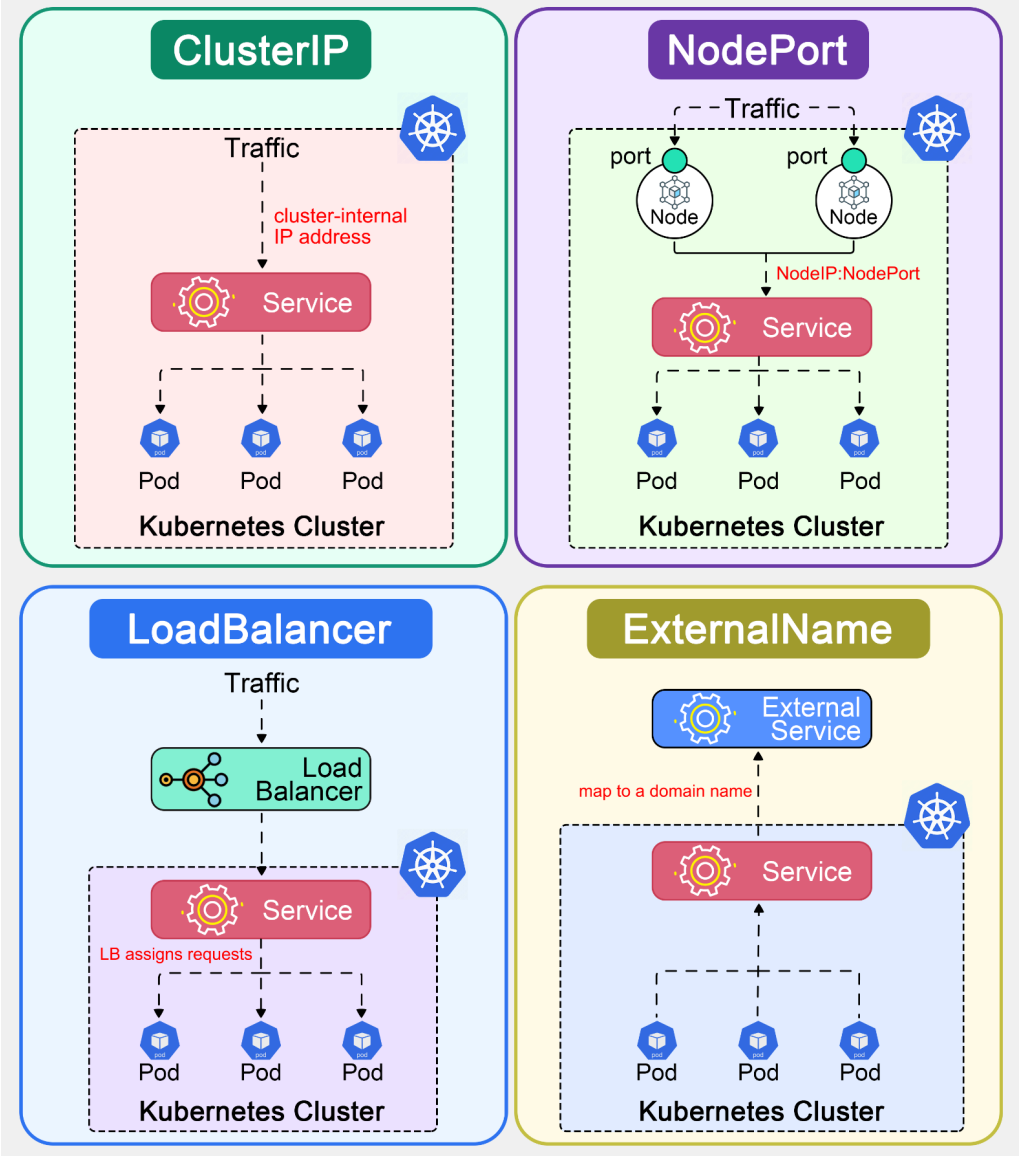
[1] Kubernetes
[2] Deploy
[3] Kubernetes Cluster
[4] Pod
[5] Desired State

Service

IP پادها را چک کنید. همانطور که می‌بینید، هر پادی آدرس متفاوت و مخصوص به خود را دارد. در این صورت مشخص نیست برای دسترسی به برنامه از کدام یک از این آدرس‌ها باید استفاده کنیم. به عنوان مثال در برنامه شما، پادهای مرتبط با سرور چگونه آدرس پادهای الستیک را پیدا کنند تا بتوانند به آن متصل شوند؟! همچنین با ساخته شدن مجدد یکی از پادها این آدرس‌ها ممکن است تغییر کند. با استفاده از دستور kubectl delete pod POD-NAME می‌توانید یکی از پادها را دیلیت کنید و پس از بالا آمدن خودکار آن به صورت مجدد توسط کوبرنتیز، IP پاد جدید را با پادی که دیلیت کرده‌اید مقایسه کنید و این مورد را تست نمایید.

Service یکی از مؤلفه‌های کوبرنتیز است که وظایف متفاوتی مانند توزیع بار و تخصیص ip مشترک به پادهای یک برنامه به منظور دسترسی به آن ها را به عهده دارد.

در شکل زیر می‌توانید Service Typeهای مختلف را مشاهده نمایید.



شما باید برای هر کدام از مؤلفه‌های نرم افزار خود یعنی API Server، Elasticsearch و Redis یک فایل Service با پسوند yaml. بنویسید. در فایلی که برای هرکدام از این سرویس‌ها می‌نویسید، مشخص می‌کنید که آن سرویس برای دسترس پذیر کردن چه پادهایی است (یک سرویس برای API Server، یک سرویس برای Elasticsearch و یک سرویس برای Redis می‌نویسید). می‌توانید نمونه‌ای از این فایل Service را در اینجا مشاهده نمایید.

ConfigMap

این شیء کوبرنتیز به شکل لیستی از کلید-مقدار[6] است و برای ذخیره کردن اطلاعات غیر محرمانه[7] (به عنوان مثال اطلاعاتی غیر از password) به کار می‌رود. استفاده از ConfigMap باعث می‌شود که تنظیمات برنامه نیازی به hard code شدن در خود کد نداشته باشند؛ در نتیجه configهای مرتبط با اپلیکیشن و environment، از image شما جدا می‌شود و اپلیکیشن شما portable می‌شود.

برای برنامه خود، فایل ConfigMap بنویسید.

اکنون پس از نوشتن فایل‌های بالا، در دایرکتوری‌ای که فایل‌های yaml شما قرار دارد، با استفاد از دستور `kubectl apply -f .` برنامه خود را بر روی کلاستر کوبرنتیز اعمال نمایید.

موارد زیر را در فایل گزارش خود نمایش دهید:

- با استفاده از دستور `kubectl get pods` پادهای بالا آمده را نمایش دهید. نام پادها بر اساس کدام فیلد از فایل Deployment تعیین شده است؟
- نشان دهید که آدرس IP پادها متفاوت هستند و در نتیجه تعریف Service الزامی است.
- یکی از مزایای Service تعیین یک IP ثابت و یکتا برای دسترسی به پادهای یک برنامه است. این کار توسط کدام یک از Service Typeها انجام می‌پذیرد. (با توجه به اینکه انواع مختلف سرویس وجود دارد، چرا الزامی به مشخص کردن نوع این سرویس خاص در فایل Service وجود ندارد؟)
- آدرس port و targetPort سرویس که بر روی کلاستر دیپلوی شده است را نشان دهید.

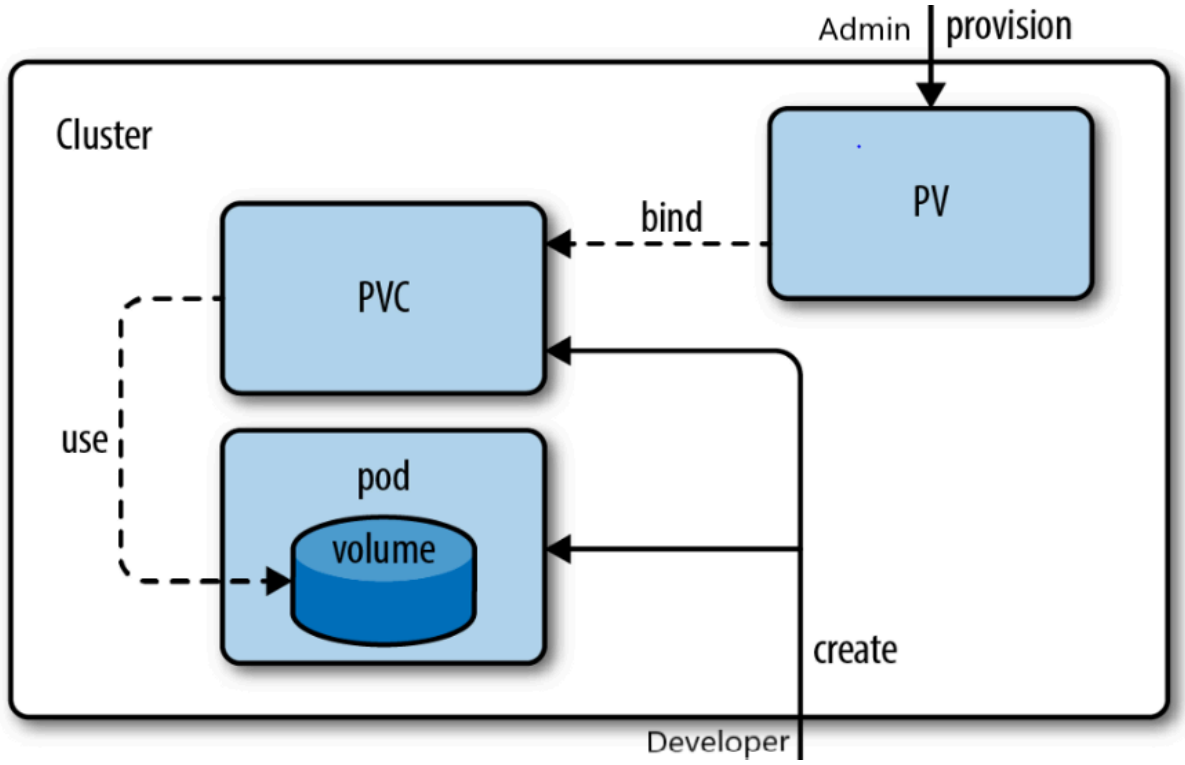
[6] Key-Value
[7] Non-Confidential

بخش ۲

PV & PVC

در ابتدا نشان دهید که با از بین رفتن پاد Redis، اطلاعات cache شده نیز از بین می‌روند. (این مورد را در فایل گزارش خود نمایش دهید)

مشکل از بین رفتن محتوای cache ردیس با پایین آمدن دیتابیس، به کمک تعریف Persistent Volume و Persistent Volume Claim حل خواهد شد. Persistent Volume یا به اختصار PV حافظه‌هایی در سطح کلاستر و مستقل از پادها می‌باشند. این حافظه‌ها می توانند به صورت های گوناگونی مثل بخشی از حافظه سیستم یا در فضای ابری وجود داشته باشند. معمولا در واقعیت ادمین یک کلاستر PVهایی با حافظه‌ها و ویژگی‌های گوناگون را تعریف می‌کند. کاربران با تعریف یک Persistent Volume Claim و دیپلوی کردن آن بر روی کلاستر می‌توانند درخواست خود برای دریافت PV را به کوبرنتیز معرفی کنند. جزئی به نام PersistentVolumeController در Kubernetes Control Plane اقدام به اختصاص volume به claimها (درخواست ها) می‌کند. درصورتی یک claim به یک volume، bind می‌شود که میزان حافظه‌ی volume از میزان حافظه‌ای که claim نیاز دارد بیشتر باشد و به طور کلی نیازهای آن claim را بتواند برآورده کند. اگر چنین volumeای یافت نشود، claim به حالت pending خواهد رفت. (اگر Dynamic Provisioning فعال باشد، درصورتی که یک claim از سمت کاربران کلاستر درخواست شود و volume مناسب برای آن پیدا نشود، یک volume برای آن ساخته می‌شود)



برای cache ردیس PV و PVC بنویسید و آنها را با استفاده از `kubectl apply -f` بر روی کلاستر کوبرنتیز اعمال نمایید. نمونه ای از PV و PVC

اکنون با استفاده از دستور `kubectl get pv` و `kubectl get pvc` می‌توانید وضعیت PVها و PVCها را مشاهده نمایید. اکنون نشان دهید که با از بین رفتن پاد Redis، اطلاعات آن از بین نمی‌روند. (این مورد را در فایل گزارش خود نمایش دهید)

بخش ۳

اکنون می‌خواهیم عملکرد کل سیستم را بر روی کلاستر کوبرنتیز آزمایش نماییم و از صحت آن اطمینان حاصل نماییم. برای این کار می‌توانید یکی از پورت های localhost را بر روی پورت مورد نظر از پادتان به کمک دستور زیر مرتبط کنید و سپس به آن درخواست بزنید:

```
kubectl port-forward {YOUR_PODNAME} local_port_num:pod_port_num
```

همچنین می‌توانید از روش‌های دلخواه دیگر برای این کار استفاده نمایید.

موارد زیر را در فایل گزارش خود نمایش دهید:

- تصویری از صحت پاسخ‌دهی سیستم در گزارش قرار دهید.
- نشان دهید توزیع بار بر روی پادهای مختلف سرور به درستی صورت می‌پذیرد. روش انجام این کار دلخواه است. ایده پیشنهادی: در خروجی سرور نام پاد را نیز نمایش دهید.

نکات مربوط به تحویل تمرین

- تمرین دارای تحویل آنلاین می‌باشد. از استفاده از کدهایی که توانایی توضیح آنها را ندارید بپرهیزید!
- سوالات خود را می‌توانید با تدریسیاران مرتبط مطرح نمایید.
- هرگونه تقلب باعث صفر شدن طرفین می‌شود.

مواردی که باید ارسال شوند:

یک فایل زیپ با نام **StudentID_HW2KubeApp.zip** که شامل موارد زیر می‌باشد (هر کدام از موارد را در پوشه‌های جداگانه قرار دهید)

- فایل‌های yaml نوشته شده
- فایل‌های مربوط به اپلیکیشن خود نظیر کد آن، Dockerfile و ... در صورتی که به نظرتان لازم است
- گزارشی که حداقل باید شامل موارد مطرح شده در توضیحات تمرین (به همراه اسکرین شات) باشد

موفق باشید

تیم تدریسیاری درس مبانی رایانش ابری