



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)
دانشکده مهندسی کامپیوتر



درس رایانش ابری

(پاییز ۱۴۰۲)

تمرین دوم (بخش دوم)

کوبرنتیز

طراحی تمرین:

خانم شریفی و خانم اویارحسینی

استاد درس:

آقای دکتر جوادی

نکات تمرین

- سوالات خود را می‌توانید با تدریس‌یاران مرتبط مطرح کنید.
 - پاسخ تمرین را در قالب یک فایل زیپ با فرمت SID_HW۲.zip ارسال بفرمایید.
 - لطفاً پاسخ خود شامل بخش اول و دوم تمرین را تا تاریخ **۱۴۰۲ / آبان / ۲۶** در سامانه بارگذاری فرمایید.
 - پاسخ به تمرین‌ها انفرادی است. از استفاده از کدهایی که بر آن‌ها مسلط نیستید پرهیزید.
 - تمرین تحویل آنلاین خواهد داشت.
 - مواردی که باید برای تمرین ارسال شوند به شرح زیر می‌باشند: (گزارش دو بخش می‌تواند یک فایل یا دو فایل جداگانه باشد).
 - تمام موارد خواسته شده در بخش اول تمرین
 - فایل‌های Service، Deployment، ConfigMap و مربوط به سرور پیش‌بینی هوا
 - فایل‌های Service، Deployment، ConfigMap، PV، PVC و مربوط به دیتابیس ردیس
 - اجرای تست جهت بررسی صحت عملکرد سیستم.
 - گزارشی شامل مواردی که در قسمت گزارش مراحل آمده است.
 - برای شفاف‌تر شدن نحوه حل تمرین می‌توانید از دو کارگاه ضبط شده **داکر** و **کوبرنتیز** کمک بگیرید.
- (این کارگاه‌ها توسط آقای سامان حسینی ارائه شده‌اند.)

مراحل انجام تمرین

به منظور دسترسی سریع تر، تمام فعالیت های تمرین به همراه موارد مورد نیاز برای گزارش در زیر آورده شده است. توضیحات تکمیلی صرفا به منظور درک مطالب و حل راحت تر سوال ها ذکر شده است. در صورت تسلط می توانید از مطالعه آن ها صرف نظر کنید.

۱. Minikube را بر روی سیستم خود اجرا کنید. [موارد گزارش ۱](#)

۲. با تعریف سه فایل Service، Deployment و ConfigMap، برنامه خود را بر روی کلاستر کوبرنتیز اعمال کنید.

[موارد گزارش ۲](#)

۳. دیتابیس ردیس را بر روی کلاستر اعمال کنید. [موارد گزارش ۳](#)

۴. مشکل از دست رفتن اطلاعات cache پس از پایین آمدن pod ردیس را برطرف کنید. [موارد گزارش ۴](#)

۵. صحت عملکرد سیستم را به وسیله ارسال درخواست به سرور هواشناسی از طریق یک پاد در کلاستر آزمایش کنید.

[موارد گزارش ۵](#)

توضیحات تکمیلی

۱. در این بخش هدف کار با ابزاری است که با آن می توانید یک کلاستر کوبرنتیز را به صورت ساده و سریع بر روی سیستم خود بالا بیاورید. این ابزار Minikube نام دارد. روش های مختلفی برای اجرای این ابزار پس از نصب وجود دارد. [اینجا](#) می توانید لیست کامل درایورهای Minikube و روش بالا آوردن آن برای هر سیستم عامل را مشاهده کنید. استفاده از هر یک از این درایورها برای بالا آوردن Minikube مجاز می باشد. تعدادی از درایورهای پیشنهادی عبارتند از :

- VirtualBox: درایور پیش فرض است. برای استفاده از این درایور باید VirtualBox را روی سیستم خود داشته باشید.
- Docker: ساده ترین درایور برای استفاده استفاده از داکر است که در آن Minikube به صورت یک کانتینر بالا می آید.
- QEMU: با کمک این شبیه ساز، Minikube به صورت یک ماشین مجازی بالا می آید.

گزارش ۱:

A. نشان دهید این مرحله با موفقیت انجام شده است.

۲. پس از تعریف سه فایل زیر، با استفاده از دستور `kubectl apply -f` برنامه خود را بر روی کلاستر اعمال کنید.

• Deployment

برای دیپلوی کردن یک برنامه بر روی کوبرنتیز شما می توانید مستقیماً از دستوراتی مثل `kubectl run` استفاده کنید. این دستور مستقیماً یک پاد بر روی کلاستر کوبرنتیز می سازد ([اطلاعات بیشتر](#)). اما اعمال کردن مستقیم یک برنامه بر روی کلاستر از اهداف اصلی کوبرنتیز مثل `auto scaling` پشتیبانی نمی کند. 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 بر روی کلاستر کوبرنتیز استفاده می شود.

برای دیپلوی کردن یک برنامه بر روی کلاستر کوبرنتیز از یک فایل به نام `Deployment` استفاده می کنیم. این فایل شرایط مطلوب ما برای برنامه را در اختیار کوبرنتیز قرار می دهد و `Kubernetes control plane` تضمین می کند که وضعیت کلاستر همیشه به همین شکل باقی بماند. مثلاً 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 می سازد تا تعداد `replica` ها سه تا باقی بماند.

یک فایل به نام `server-depoyment.yml` و یا هر نام دیگری با پسوند `yml`. بسازید و اطلاعات `Deployment` خود را در آن وارد کنید. (تعداد `replica` را ۳ قرار دهید).

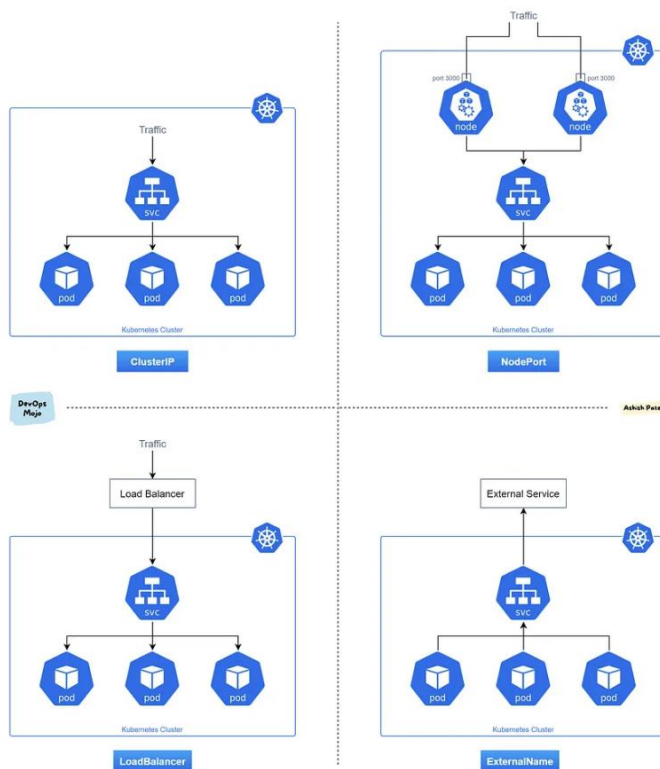
نمونه‌ای از فایل دیپلویمنت را می‌توانید در [اینجا](#) ببینید. (قسمت `ConfiMap` نیاز به افزوده شدن دارد).
فیلدهای فایل:

`Kind` در ابتدای فایل اشیا^۱ کوبرنتیز وجود دارد. و مشخص می‌کند این شی از چه نوعی است.
با استفاده از فیلد `image` کوبرنتیز قادر است این `image` را از `dockerhub` دریافت کند.

• Service

`ip` پادها را چک کنید. همانطور که می‌بینید، پادها آدرس‌های متفاوتی دارند. در این صورت مشخص نیست برای دسترسی به برنامه از کدام یک از این آدرس‌ها باید استفاده کنیم زیرا همه‌ی پادها `image` یکسان دارند. و نیز با ساخته شدن مجدد یکی از پادها این آدرس‌ها ممکن است تغییر کند. با استفاده از دستور `kubectl delete pod [POD_NAME]` می‌توانید این مورد را تست کنید.

`Service` یک شی کوبرنتیز است که وظایف متفاوتی مثل توزیع بار^۲ و تخصیص `ip` مشترک به پادهای یک برنامه به منظور دسترسی به آن‌ها را به عهده دارد. انواع سرویس‌ها در شکل زیر وجود دارد.



^۱ Objects

^۲ Load Balancing

برای برنامه خود فایل Service بنویسید. نمونه‌ای از فایل Service در این [لینک](#) موجود است.

• ConfigMap

این شی کوبرنتیز به شکل لیستی از کلید و مقادیر است و باعث می‌شود که تنظیمات برنامه نیازی به هاردکد شدن در خود کد نداشته باشند.

برای برنامه خود، فایل ConfigMap را بنویسید. (کلید و مقادیر این فایل مواردی است که در بخش اول تمرین به عنوان کانفیگ تعریف کرده‌اید).

گزارش ۲:

A. با استفاده از دستور `kubectl get pods` نام پادهای ساخته‌شده را بیابید و آن‌ها را نمایش دهید. این نام‌ها براساس کدام فیلد از فایل Deployment تعیین شده است؟

B. نشان دهید که آدرس ip پادها متفاوت هستند و در نتیجه تعریف Service الزامی است.

C. یکی از مزایای سرویس تعیین یک ip ثابت و یکتا برای دسترسی به پادهای یک برنامه است. این کار توسط کدام یک از انواع سرویس 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؟)

D. آدرس port و targetPort سرویس که بر روی کلاستر دیپلوی شده است را نشان دهید.

۳. مشابه با قسمت ۲، فایل‌های مورد نیاز را برای ردیس تعریف و نهایتاً دیتابیس را بر روی کلاستر کوبرنتیز اعمال کنید.

گزارش ۳:

A. نشان دهید با از بین رفتن pod دیتابیس، اطلاعات cache شده نیز از بین خواهد رفت.

۴. مشکل از بین رفتن محتوای cache ردیس با پایین آمدن دیتابیس به کمک تعریف Persistent و Persistent Volume حل خواهد شد.

Persistent Volume یا به اختصار PV حافظه‌هایی در سطح کلاستر و مستقل از پادها می‌باشند. این حافظه‌ها می‌توانند به صورت‌های گوناگونی مثل بخشی از حافظه سیستم یا در فضای ابری وجود داشته باشند. معمولاً در واقعیت مدیر 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 (درخواست‌ها) می‌کند. و منطقاً در صورتی یک claim به volume bind می‌شود که میزان حافظه‌ی volume از میزان حافظه‌ای که claim نیاز دارد بیشتر باشد. اگر چنین volume‌ای یافت نشود، claim به حالت pending خواهد رفت. (اگر [Dynamic provisioning](#) فعال باشد، در صورتی 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 VP و VPC بنویسید و آن‌ها را با استفاده از `kubectl apply -f` بر روی کلاستر کوبرنتیز اعمال کنید. [مثال VP و مثال VPC](#)

گزارش ۴ :

A. نشان دهید که آیا minikube، به عنوان یک ابزار نصب کوبرنتیز dynamic provisioning را به صورت پیش فرض فعال می‌کند یا خیر. (امتیازی)

(ایده: یک vpc با حجم مورد نیاز فرضی k بر روی کلاستر دیپلوی کنید. و چند vpc با حجمی کمتر از k نیز بسازید. بررسی کنید که آیا وضعیت claim، pending شده است یا خیر) با استفاده از دستور `kubectl get pvc` می‌توانید وضعیت claim‌های موجود را مشاهده کنید.

B. آیا در صورتی که یک volume حجم کافی داشته باشد می‌توان آن را به بیش از یک claim اختصاص داد ؟ یک مزیت برای این سیاست را بنویسید. (امتیازی)

۵. می‌خواهیم کل سیستم را بر روی کلاستر کوبرنتیز آزمایش کنیم. سرویس‌های تعریف شده از نوع Cluster IP service هستند. IP این نوع از سرویس‌ها صرفاً در داخل کلاستر قابل دسترسی است. بنابراین برای تست سیستم نیاز به پادی داریم که در داخل کلاستر باشد.

در بخش اول تمرین ۲ یک image حاوی curl ساخته بودیم.

به کمک دستور `kubectl run` که به صورت زیر می‌باشد:

```
kubectl run NAME --image=image [--env="key=value"] [--port=port] [--replicas=replicas]
[--dry-run=bool] [--overrides=inline-json] [--command] -- [COMMAND] [args...]
```

Image مربوط به curl را دیپلوی کنید و با استفاده از curl به IP سرویس درخواست بزنید و صحت عملکرد کل سیستم را نشان دهید. به جای استفاده از image مربوط به curl می‌تواندی از روی سیستم خودتان درخواست به سرور بزنید. برای این کار لازم است یکی از پورت‌های local host را بر روی پورت مورد نظر از پادتان به کمک دستور زیر مرتبط کنید:

```
kubectl port-forward your_pod local_port_num:pod_port_num
```

گزارش ۵:

A. تصویری از صحت پاسخ‌دهی سیستم در گزارش قرار دهید.

B. نشان دهید توزیع بار بر روی پادهای مختلف سرور به درستی صورت می‌پذیرد. روش انجام این کار دلخواه است. ایده پیشنهادی: در خروجی سرور هواشناسی نام پاد را نیز نمایش دهید.

پیروز باشید.

تیم تدریسیاری درس رایانش ابری