گزارش تمرین دوم رایانش ابری

آراد فیروزکوهی - ۹۸۳۱۰۴۷

گام اول: یک Dockerfile ساده می‌سازیم که به طور دیفالت طبق خواسته گوگل را کرل میکند. در صورت نیاز به کرل جای دیگر، با docker exec می‌توان آن را انجام داد. طبق اسکرین شات آن را بیلد و پوش می‌کنیم.

```
~/PycharmProjects/CCHW2_9831047/Part1
❯ docker build . -t aradfir/curler
Sending build context to Docker daemon  2.048kB
Step 1/2 : FROM alpine/curl
 ---> 6c746001fe6d
Step 2/2 : CMD ["curl","www.google.com/"]
 ---> Using cache
 ---> 3788b0cc67ef
Successfully built 3788b0cc67ef
Successfully tagged aradfir/curler:latest
~/PycharmProjects/CCHW2_9831047/Part1
❯ docker push aradfir/curler
Using default tag: latest
The push refers to repository [docker.io/aradfir/curler]
bb31d64dd136: Layer already exists
8cdeb0677eff: Layer already exists
8d3ac3489996: Layer already exists
latest: digest: sha256:6d0baf038033ad774b47862f02859d9327f1251efc0c594e1c404bd952776e15 size: 946
~/PycharmProjects/CCHW2_9831047/Part1
❯ |
```

اجرای آن:

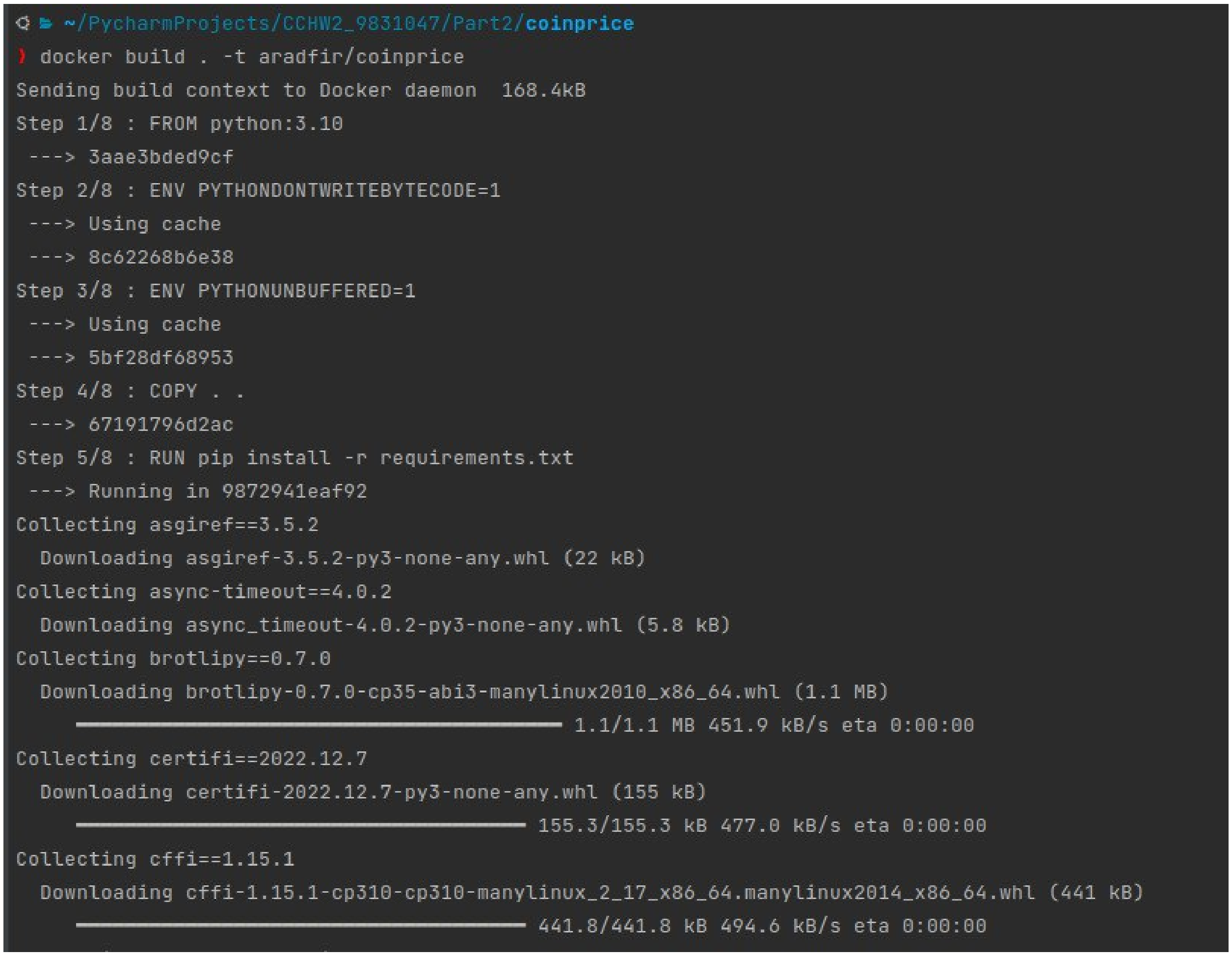
```
~/PycharmProjects/CCHW2_9831047/Part1                                                        ✘ INT ✚ cchw2 ⊙ 23:45:16
❯ docker run aradfir/curler:latest
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0<!doctype html><html itemscope="" itemtype="http://schema.org/WebPage" lang="uk"><head><meta content="tex
t/html; charset=UTF-8" http-equiv="Content-Type"><meta content="/images/branding/googleg/1x/googleg_standard_color_128dp.png" itemprop="image"><title>Google</title><script nonce="B4TM
cMYV-tUuTbo5fpr1cQ">(function(){window.google={kEI:'YGmjY4q5DLWEhbIPmJykkA0',kEXPI:'0,18167,1341242,6058,207,2415,2389,2316,383,246,5,1129120,1197720,681,302561,77529,16114,28684,2243
1,1361,12319,17580,4998,13228,3847,6885,3737,22741,5081,885,708,1279,2742,149,1103,840,1987,4310,3514,606,2024,2296,14670,3227,2845,7,5599,23475,4696,1851,2614,13142,3,346,230,6460,14
8,13975,4,1528,2304,7039,27731,7357,13658,4437,16786,5830,2527,4094,4052,3,3541,1,42154,2,14022,25739,5679,1020,2381,28742,4567,6259,23418,1252,5835,14968,4332,22,7462,445,2,2,1,26632
,8155,6680,701,1401,14569,872,19634,7,1922,5784,3995,21390,389,9543,4832,23049,3454,20137,14,82,3890,751,14144,742,679,1622,1779,2224,1627,1125,8836,154,5901,753,989,813,86,1428,2306,
44,1497,568,716,272,2094,172,179,90,399,96,426,848,186,42,291,2669,723,443,482,470,941,20,3,261,606,2261,289,135,1304,246,951,2219,947,556,4,6,291,301,525,204,2642,71,3,538,599,551,48
4,601,267,764,580,80,4,1001,166,886,200,5,342,148,1029,130,108,96,366,218,85,238,114,11,78,35,333,174,266,1599,129,340,500,62,1518,1111,1015,494,62,1123,76,5,97,37,165,261,36,12,105,1
91,489,692,293,2,237,889,491,560,259,1453,3075,770,117,111,4,19,38,40,2,355,17,182,86,711,976,417,655,619,70,1455,5275939,12,39,8799141,3311,141,795,19735,1,1,346,3570,9,2,6,14,2,1,18
,28,13,21,23946800,4042143,1964,14297,2376,3405,5595,11,3834,6970,227',kBL:'Dl4b'};google.sn='webhp';google.kHL='uk';})();(function(){
var f=this||self;var h,k=[];function l(a){for(var b;a&&(!a.getAttribute||!(b=a.getAttribute("eid")));)a=a.parentNode;return b||h}function m(a){for(var b=null;a&&(!a.getAttribute||!(b=
a.getAttribute("leid")));)a=a.parentNode;return b}
function n(a,b,c,d,g){var e="";c||-1!==b.search("&ei=")||(e="&ei="+l(d),-1===b.search("&lei=")&&(d=m(d))&&(e+="&lei="+d));d="";!c&&f._cshid&&-1===b.search("&cshid=")&&"slh"!==a&&(d="&
cshid="+f._cshid);c=c||"/"+(g||"gen_204")+"?atyp=i&ct="+a+"&cad="+b+e+"&zx="+Date.now()+d;/^http:/i.test(c)&&"https:"===window.location.protocol&&(google.ml&&google.ml(Error("a"),!1,{
src:c,glmm:1}),c="");return c};h=google.kEI;google.getEI=l;google.getLEI=m;google.ml=function(){return null};google.log=function(a,b,c,d,g){if(c=n(a,b,c,d,g)){a=new Image;var e=k.leng
th;k[e]=a;a.onerror=a.onload=a.onabort=function(){delete k[e]};a.src=c}};google.logUrl=n;}).call(this);(function(){google.y={};google.sy=[];google.x=function(a,b){if(a)var c=a.id;else
```

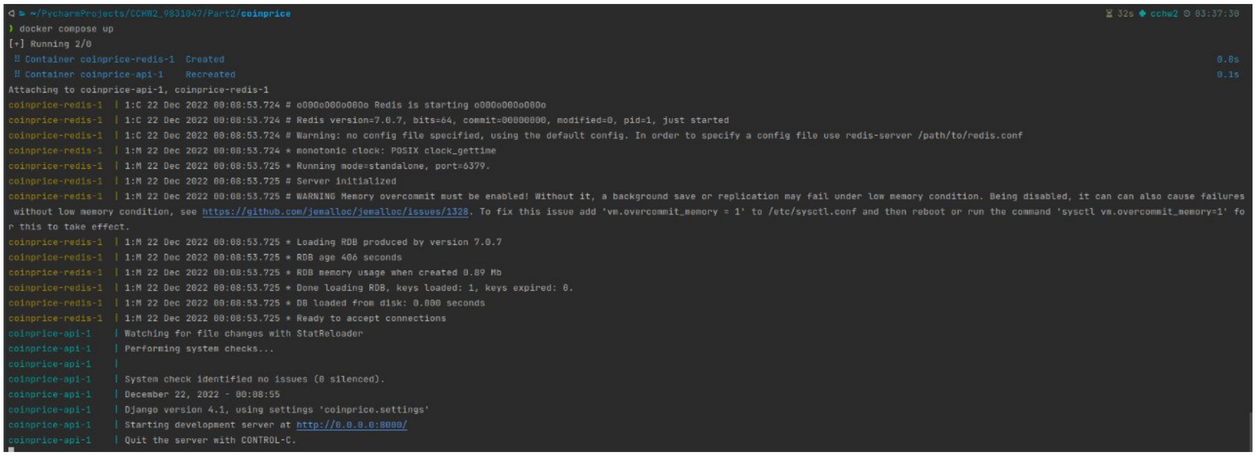
```
~/PycharmProjects/CCHW2_9831047/Part1
❯ docker run -d aradfir/curler:latest sleep infinity
9d232a346bd4853842a5259114a7ca234c0beae0107b0ecf54643d83fdb56bf2
~/PycharmProjects/CCHW2_9831047/Part1
❯ docker exec 9d232a346bd4 curl google.com
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   219  100   219    0     0    290      0 --:--:-- --:--:-- --:--:--   290<HTML><HEAD><meta http-equiv="content-type" content="text/html;charset=utf-8">
<TITLE>301 Moved</TITLE></HEAD><BODY>
<H1>301 Moved</H1>
The document has moved
<A HREF="http://www.google.com/">here</A>.
</BODY></HTML>
100   219  100   219    0     0    290      0 --:--:-- --:--:-- --:--:--   290
~/PycharmProjects/CCHW2_9831047/Part1
❯ |
```

برای بخش دوم:

یک پروژه جنگو پیاده‌سازی شده که از coingecko قیمت را خوانده و آن را در کش می‌ریزد. کنار آن یک فایل env. گذاشته شده که متغییر هایی همچون پورت جنگو، پورت ردیس، توکنی که دیفالت هست و اندپوینت API سایت در آن ثبت شده. سپس با کمک یک docker-compose ردیس و ایمج خود را با هم بالا آورده و یک volume و network به آن می‌دهیم.

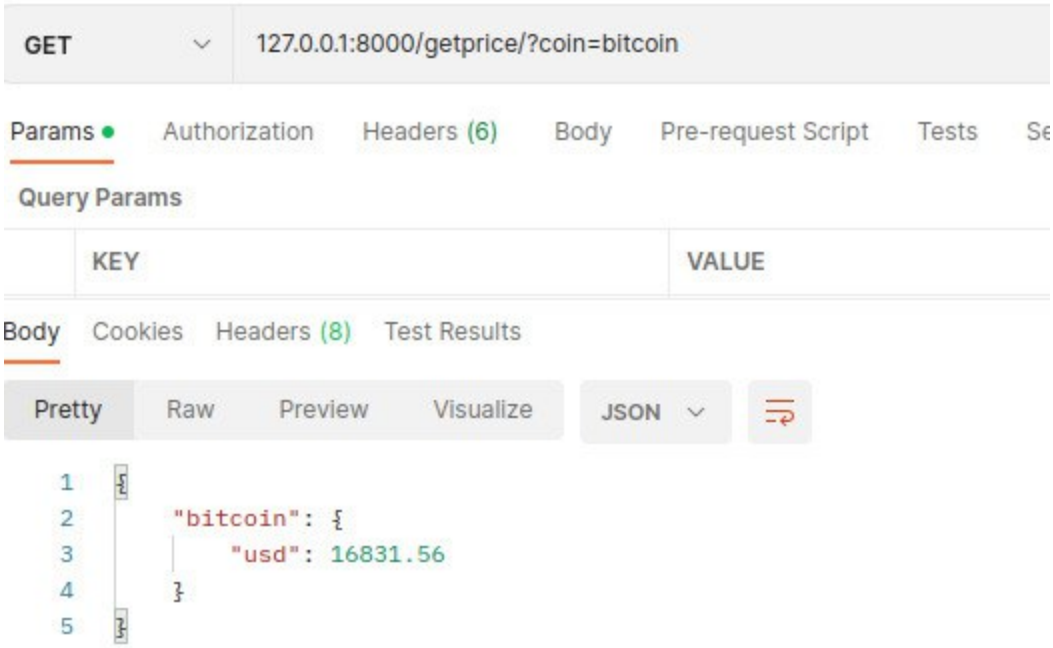
پس در مجموع: Dockerfile را بیلد و پوش می‌کنیم، سپس docker compose up می‌زنیم.





همانطور که در اسکرین‌شات فوق مشهود است، ردیس هنگام بالا آمدن از volume خود فایل را هم خواند و حالت persistent رعایت شده.

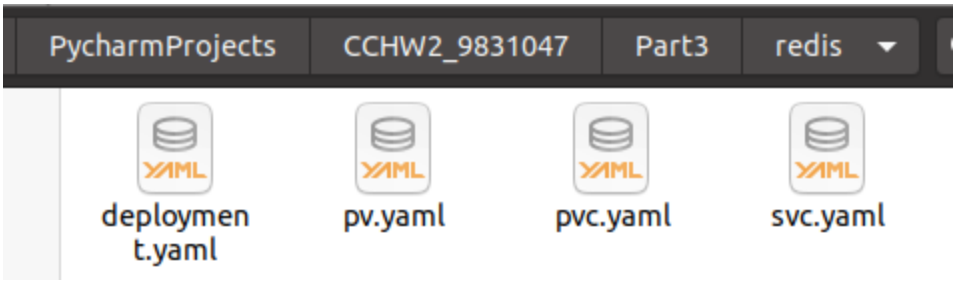
نمونه ریکوئست را در تصویر زیر میبینیم:

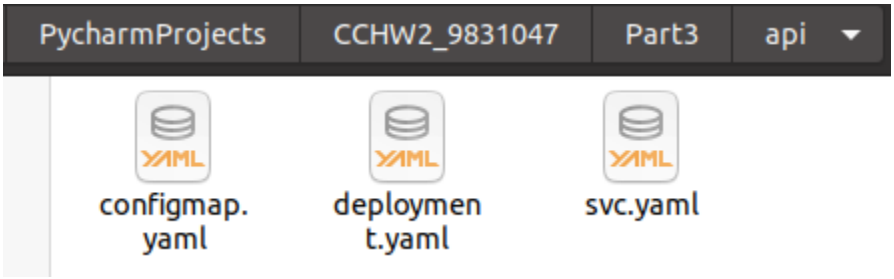



```
coinprice-api-1    | December 22, 2022 - 00:08:55
coinprice-api-1    | Django version 4.1, using settings 'coinprice.settings'
coinprice-api-1    | Starting development server at http://0.0.0.0:8000/
coinprice-api-1    | Quit the server with CONTROL-C.
coinprice-api-1    | reading from cache! 1214
coinprice-api-1    | [22/Dec/2022 00:09:28] "GET /getprice/?coin=ethereum HTTP/1.1" 200 30
coinprice-redis-1  | 1:M 22 Dec 2022 00:09:40.777 * DB saved on disk
coinprice-api-1    | [22/Dec/2022 00:09:40] "GET /getprice/?coin=bitcoin HTTP/1.1" 200 30
coinprice-api-1    | reading from cache! 16831
coinprice-api-1    | [22/Dec/2022 00:09:50] "GET /getprice/?coin=bitcoin HTTP/1.1" 200 30
```

ریکوئست اول برای اتریوم است که همان مقدار در کش بود.
در فراخوانی دوم که برای بیتکوین هست، ابتدا کش آپدیت شده و مقدار روی دیسک می‌رود، سپس پاسخ داده می‌شود. در فراخوانی سوم که باز برای بیت‌کوین هست، این بار مقدار از کش خوانده شده.

بخش سوم:
پس از نصب minikube و kubectl و اجرای minikube start، کلاستر k8s ما آماده است.
سپس با کمک تمپلیت‌های موجود در اینترنت و تغییر برای نیازهای ما، منیفست‌هایی برای سرویس، دیپلویمنت و کانفیگ‌مپ api و سپس pvc ،pv، و سرویس و دیپلویمنت برای ردیس می‌نویسیم.





در نهایت در این دو پوشه، k apply -f . میزنیم تا تغییرات روی کلاستر اعمال شوند.

```
❯ k get all
NAME                          READY   STATUS    RESTARTS   AGE
pod/api-848b65fb5b-8ctbz      1/1     Running   0          48m
pod/api-848b65fb5b-srb4q      1/1     Running   0          48m
pod/redis-b6b8f566b-w5v9q     1/1     Running   0          49m

NAME                 TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)    AGE
service/api          ClusterIP   10.111.53.193   <none>        8000/TCP   49m
service/kubernetes   ClusterIP   10.96.0.1       <none>        443/TCP    3d17h
service/redis        ClusterIP   10.108.235.40   <none>        6379/TCP   49m

NAME                     READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/api      2/2     2            2           48m
deployment.apps/redis    1/1     1            1           49m

NAME                                DESIRED   CURRENT   READY   AGE
replicaset.apps/api-848b65fb5b      2         2         2       48m
replicaset.apps/redis-b6b8f566b     1         1         1       49m
```

بخش چهارم:

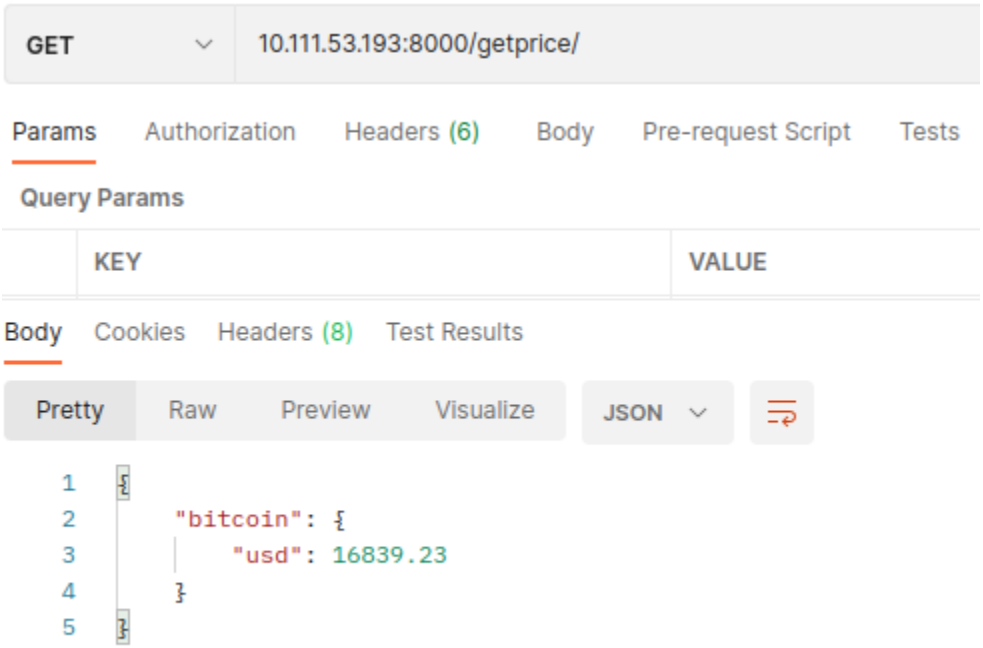
با دستور زیر یک پاد ساخته و در آن دستور sh اجرا می‌کنیم. از آنجا که درون کلاستر، سرویس‌ها همدیگر را با پترن namespace.service:port پیدا می کنند، می‌توانیم از دی‌ان‌اس داخلی کوبرنتیز استفاده کنیم. از آنجا که پاد جدید و پادهای api در یک نیم‌اسپیس اند، این بخش از URI حذف می‌شود.

```
❯ k run -it --rm curl --image aradfir/curler --image-pull-policy=Never -- sh
If you don't see a command prompt, try pressing enter.
/ # curl api:8000/getprice/
{"bitcoin": {"usd": 16839.23}}/ #
/ #
```

توضیح دستور: با run یک پاد جدید ساخته میشود. با -it حالت تعاملی میگیرد و ورودی خروجی میگیرد. با –rm پس از خروج از شل، پاد پاک می‌شود. نام این پاد curl است و از ایمج ما استفاده میکند. در نهایت ذکر شده که از ایمج درون کامپیوتر استفاده شود و شل را اجرا کند.

برای استفاده راحت تر از کلاستر می‌توان آدرس سرویس که کار لود بالانسینگ را هم انجام می‌دهد را به کمک k get services می‌گیریم و مشابه با قبل در پست‌من از آن استفاده می‌کنیم.

```
❯ kubectl get svc
NAME         TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)    AGE
api          ClusterIP   10.111.53.193   <none>        8000/TCP   57m
kubernetes   ClusterIP   10.96.0.1       <none>        443/TCP    3d17h
redis        ClusterIP   10.108.235.40   <none>        6379/TCP   57m
```



برای استفاده حتی راحت تر می‌توان از portforwarding استفاده کرد.

```
❯ kubectl port-forward services/api 8000:8000
Forwarding from 127.0.0.1:8000 -> 8000
Forwarding from [::1]:8000 -> 8000
```

حال سرویس ما در localhost:8000 قابل مشاهده است.

در اسکرین‌شات‌های فوق، اکثر منابع موجود در کلاستر را نظاره کردیم. سایر منابع در اسکرین‌شات زیر آمده‌اند:

```
❯ k get endpoints
NAME         ENDPOINTS                          AGE
api          172.17.0.4:8000,172.17.0.5:8000    62m
kubernetes   192.168.49.2:8443                  3d17h
redis        172.17.0.3:6379                    63m
 ~/PycharmProjects/CCHW2_9831047/Part2/coinprice
❯ k get configmaps
NAME               DATA   AGE
api                7      63m
kube-root-ca.crt   1      3d17h
```

برای مشاهده توضیع بار، می‌توان لاگ هر پاد api را دید:

```
❯ k logs api-848b65fb5b-8ctbz
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
December 22, 2022 - 02:45:55
Django version 4.1, using settings 'coinprice.settings'
Starting development server at http://0.0.0.0:8000/
Quit the server with CONTROL-C.
reading from cache! 16839
[22/Dec/2022 03:16:05] "GET /getprice/ HTTP/1.1" 200 30
reading from cache! 16839
[22/Dec/2022 03:16:07] "GET /getprice/ HTTP/1.1" 200 30
reading from cache! 16839
[22/Dec/2022 03:16:08] "GET /getprice/ HTTP/1.1" 200 30
reading from cache! 16839
[22/Dec/2022 03:16:09] "GET /getprice/ HTTP/1.1" 200 30
reading from cache! 16839
[22/Dec/2022 03:16:10] "GET /getprice/ HTTP/1.1" 200 30
reading from cache! 16839
[22/Dec/2022 03:16:11] "GET /getprice/ HTTP/1.1" 200 30
reading from cache! 16839
[22/Dec/2022 03:16:11] "GET /getprice/ HTTP/1.1" 200 30
reading from cache! 16839
[22/Dec/2022 03:37:59] "GET /getprice/ HTTP/1.1" 200 30
 ~/PycharmProjects/CCHW2_9831047/Part2/coinprice
❯ k logs api-848b65fb5b-srb4q
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
December 22, 2022 - 02:45:55
Django version 4.1, using settings 'coinprice.settings'
Starting development server at http://0.0.0.0:8000/
Quit the server with CONTROL-C.
[22/Dec/2022 02:46:16] "GET /getprice/ HTTP/1.1" 200 30
reading from cache! 16839
[22/Dec/2022 02:46:18] "GET /getprice/ HTTP/1.1" 200 30
reading from cache! 16839
[22/Dec/2022 02:46:19] "GET /getprice/ HTTP/1.1" 200 30
reading from cache! 16839
[22/Dec/2022 02:46:19] "GET /getprice/ HTTP/1.1" 200 30
reading from cache! 16839
[22/Dec/2022 03:43:52] "GET /getprice/ HTTP/1.1" 200 30
 ~/PycharmProjects/CCHW2_9831047/Part2/coinprice
❯
```

که در آن هر پاد به تعدادی درخواست پاسخ داده. همانطور که مشاهده می‌شود، هر دو از یک کش استفاده می‌کنند.