



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

دانشکده مهندسی کامپیووتر

تمرین دوم

درس رایانش ابری

هلیاسادات هاشمی‌پور - ۹۸۳۱۱۰۶

استاد

دکتر جوادی

آذر ۰۱

گام اول

• ارسال ایمیج ساخته شده بر روی داکرهاپ

ابتدا یک داکرفایل درست میکنیم که از alpine استفاده کرده و همچنین بر روی آن curl را نصب میکنیم.

```
FROM alpine
```

```
RUN apk update && apk add curl
```

```
CMD ash
```

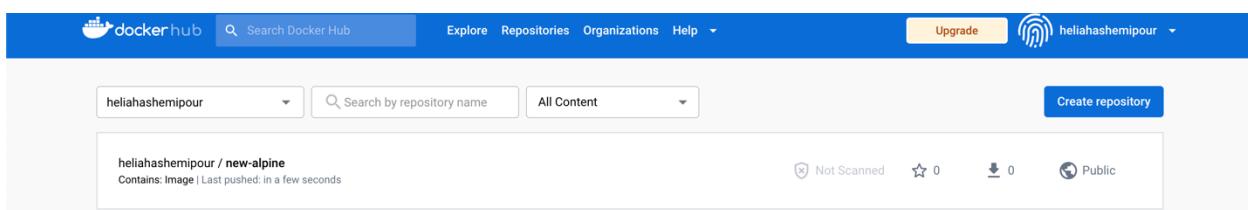
حال با استفاده از دستور زیر، ایمیج آن را تشکیل میدهیم.

```
heliaa@MacBook-Pro ~:/University/Semester7/Cloud/HW2-9831106/step1.0 | main ± docker build -t new-alpine:1.0 .
[+] Building 4.7s (7/7) FINISHED
--> [internal] load build definition from Dockerfile          0.0s
--> >> transferring dockerfile: 94B                          0.0s
--> [internal] load .dockerignore                           0.0s
--> >> transferring context: 2B                            0.0s
--> [internal] load metadata for docker.io/library/alpine:latest 4.6s
--> [auth] library/alpine:pull token for registry-1.docker.io
--> [1/2] FROM docker.io/library/alpine@sha256:8914eb54f968791faf6a863894 0.0s
--> CACHED [2/2] RUN apk update && apk add curl           0.0s
--> exporting to image                                     0.0s
--> >> exporting layers                                    0.0s
--> >> writing image sha256:40cfa75878544e2be4a35ae30f3ceea9cb6af6dfd8c8d 0.0s
--> >> naming to docker.io/library/new-alpine:1.0          0.0s
```

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them

سپس برای اینکه این ایمیج را در داکرهاپ قرار دهیم یک تگ جدید اضافه میکنیم. در ادامه آن را push میکنیم. که دستور آن به صورت زیر میباشد.

```
heliaa@MacBook-Pro ~:/University/Semester7/Cloud/HW2-9831106/step1.0 | main ± docker tag new-alpine:1.0 heliahashemipour/new-alpine:1.0
heliaa@MacBook-Pro ~:/University/Semester7/Cloud/HW2-9831106/step1.0 | main ± docker image push heliahashemipour/new-alpine:1.0
The push refers to repository [docker.io/heliahashemipour/new-alpine]
54189791f518: Pushed
ded7a220bb05: Mounted from library/python
```



نمایش ایمیج‌های موجود بر روی سیستم خود

```
| heliaa@MacBook-Pro ~/University/Semester7/Cloud/HW2-9831106/step4 | main ± docker images
REPOSITORY      TAG      IMAGE ID   CREATED      SIZE
coin-server     1.0      3ca6b8c04975  4 hours ago  71.8MB
heliahemipour/coin-server 1.0      3ca6b8c04975  4 hours ago  71.8MB
heliahemipour/coin-app    <none>   f7baf443d9a   12 hours ago 70.3MB
heliahemipour/new-ubuntu   1.0      036b4b5a498f   12 hours ago 129MB
tiangolo/uwsgi-nginx-flask latest   9634df2b32ba   2 days ago   968MB
<none>          <none>   4ee648c5b12   4 days ago   975MB
redis           latest   29ab4501eac3   4 days ago   117MB
heliahemipour/new-alpine   1.0      40cfaf7587854  5 days ago   11.9MB
new-alpine       1.0      40cfaf7587854  5 days ago   11.9MB
kicbase/stable   v0.0.36  866c1fe4e3f2  8 weeks ago  1.11GB
```

دربیافت ایمیج ساخته شده از داکرهاب

برای تست دریافت ایمیج ساخته شده در داکرهاب، ابتدا ایمیج را از سیستم خود پاک می‌کنیم (در واقع هر دو ایمیج را حذف می‌کنیم).
در ادامه ایمیج را از داکرهاب دریافت می‌کنیم.

```
| heliaa@MacBook-Pro ~/University/Semester7/Cloud/HW2-9831106/step4 | main ± docker rmi heliahemipour/new-alpine
Error: No such image: heliahemipour/new-alpine
| heliaa@MacBook-Pro ~/University/Semester7/Cloud/HW2-9831106/step4 | main ± docker rmi heliahemipour/new-alpine:1.0
Untagged: heliahemipour/new-alpine@sha256:9b22b8e4fc9e7629971a588b0d179744e0764c7452cc1e459d6194f75d6ba01
| heliaa@MacBook-Pro ~/University/Semester7/Cloud/HW2-9831106/step4 | main ± docker rmi new-alpine:1.0
Untagged: new-alpine@sha256:40cfaf758785442be4a35ae30f3ceea9c6a6fd8c8df7845650d41b817ab6
Deleted: sha256:40cfaf758785442be4a35ae30f3ceea9c6a6fd8c8df7845650d41b817ab6
| heliaa@MacBook-Pro ~/University/Semester7/Cloud/HW2-9831106/step4 | main ± docker images
REPOSITORY      TAG      IMAGE ID   CREATED      SIZE
heliahemipour/coin-server 1.0      3ca6b8c04975  4 hours ago  71.8MB
coin-server     1.0      3ca6b8c04975  4 hours ago  71.8MB
heliahemipour/coin-app    <none>   f7baf443d9a   12 hours ago 70.3MB
heliahemipour/new-ubuntu   1.0      036b4b5a498f   12 hours ago 129MB
tiangolo/uwsgi-nginx-flask latest   9634df2b32ba   2 days ago   968MB
<none>          <none>   4ee648c5b12   4 days ago   975MB
redis           latest   29ab4501eac3   4 days ago   117MB
kicbase/stable   v0.0.36  866c1fe4e3f2  8 weeks ago  1.11GB
docker          <none>   0dfb722b2a54  3 months ago 13.4MB

| heliaa@MacBook-Pro ~/University/Semester7/Cloud/HW2-9831106/step4 | main ± docker image pull heliahemipour/new-alpine:1.0
1.0: Pulling from heliahemipour/new-alpine
c158987b0551: Already exists
9c3bc4edfb61: Already exists
Digest: sha256:9b22b8e4fc9e7629971a588b0d179744e0764c7452cc1e459d6194f75d6ba01
Status: Downloaded newer image for heliahemipour/new-alpine:1.0
docker.io/heliahemipour/new-alpine:1.0
| heliaa@MacBook-Pro ~/University/Semester7/Cloud/HW2-9831106/step4 | main ± docker images
REPOSITORY      TAG      IMAGE ID   CREATED      SIZE
heliahemipour/coin-server 1.0      3ca6b8c04975  4 hours ago  71.8MB
coin-server     1.0      3ca6b8c04975  4 hours ago  71.8MB
heliahemipour/coin-app    <none>   f7baf443d9a   12 hours ago 70.3MB
heliahemipour/new-ubuntu   1.0      036b4b5a498f   12 hours ago 129MB
tiangolo/uwsgi-nginx-flask latest   9634df2b32ba   2 days ago   968MB
<none>          <none>   4ee648c5b12   4 days ago   975MB
redis           latest   29ab4501eac3   4 days ago   117MB
heliahemipour/new-alpine   1.0      40cfaf7587854  5 days ago   11.9MB
kicbase/stable   v0.0.36  866c1fe4e3f2  8 weeks ago  1.11GB
```

ساختن کانتینر از ایمیج دریافت شده از داکرهاب و اجرا دستور curl

```
| heliaa@MacBook-Pro ~/University/Semester7/Cloud/HW2-9831106/step1.0 | main ± docker run -it --rm heliahemipour/new-alpine:1.0
/ # curl google.com
<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>
/ #
```

برای ایمیج ubuntu هم همین مراحل طی شده بود.

```
heliaha@MacBook-Pro ~ % cd University/Semester7/Cloud/HW-9&10/step1 ; ./main ; docker build -t new-ubuntu:1.0 .
[+] Building 75.3s (7/7) FINISHED
   => [internal] load build definition from Dockerfile                                0.0s
   => => transferring dockerfile: 196B                                              0.0s
   => [internal] load .dockerignore                                               0.0s
   => => transferring context: 2B                                              0.0s
   => [internal] load metadata for docker.io/library/ubuntu:20.04                  3.9s
   => [auth] library/ubuntu:pull token for registry-1.docker.io                      0.0s
   => [1/2] FROM docker.io/library/ubuntu:20.04@sha256:0e0402cd13f68137edb0          9.4s
   => => resolve docker.io/library/ubuntu:20.04@sha256:0e0402cd13f68137edb0        0.0s
   => => sha256:0e402cd13f68137edb02661d2c682f217b1442f71.4.42K / 1.42KB 0.0s
   => => sha256:0e8b7f3d6c9f72feeff114ff93ea0df2b294edfa0835 529B / 529B 0.0s
   => => sha256:d547fc1a1e6c2b08efbfba58b0c51b2797f9d7ebae0 1.46KB / 1.46KB 0.0s
   => => sha256:b846cc0181ffffcc667d9444f83788efcf13116d88 28.5MB / 28.5MB 7.4s
   => => extracting sha256:b846cc0181ffffcc667d9444f83788efcf13116d88d3088b2 1.5s
   => [2/2] RUN apt-get update && apt-get install curl --                                61.6s
   => => exporting to image                                                       0.3s
   => => exporting layers                                                       0.3s
   => => writing image sha256:036b4b5a498fd0e66f0256d7f1a32db659e40884bc623 0.0s
   => => naming to docker.io/library/new-ubuntu:1.0                            0.0s
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
heliahaMacBook-Pro:~/University/semester//Cloud/mhw-003\$	step01			main ↗
heliahashemipour/new-ubuntu	1.0	036b4b5a498f	8 minutes ago	129MB
new-ubuntu	1.0	036b4b5a498f	8 minutes ago	129MB
tiangolo/uwsgi-nginx-flask	latest	9634df2032b2	2 days ago	968MB
<none>	<none>	4eeed48c5012	3 days ago	975MB
redis	latest	29ab4501eac3	3 days ago	117MB
kicbdc/stable	v0.0.36	866c1fe4e3f2	8 weeks ago	1.11GB
docker	<none>	0fdff722b2d54	3 months ago	134MB

```
heliaha[MacBook-Pro] ~[University/Semester7/Cloud/Hw2-081108/step1] main: docker tag new-ubuntu:1.0 heliahashemipour/new-ubuntu:1.0  
heliaha[MacBook-Pro] ~[University/Semester7/Cloud/Hw2-081108/step1] main: docker image push heliahashemipour/new-ubuntu:1.0  
The push refers to repository [docker.io/heliahashemipour/new-ubuntu]  
2d5b2907bcc: Pushed  
0002c93bdb37: Mounted from library/ubuntu  
1.0: digest: sha256:e77e56a2bed143af3c3066d76d9fc3a54287a0e4d08ee71845a637afa9098c size: 740
```

Docker Hub Search Docker Hub Explore Repositories Organizations Help Upgrade heliahashemipour

heliahemipour All Content Create repository

heliahemipour / new-ubuntu Contains: Image | Last pushed: a few seconds ago Not Scanned ⭐ 0 Download Public

```
heliahemipour@MacBook-Pro ~% cd /Users/heliahashemipour/Desktop/Cloud/HW2-0931106/step1 % docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
heliahemipour/new-ubuntu 1.0 036b4b5b498f 9 minutes ago 129MB
tiangolo/uwsgi-nginx-flask latest 9634df2032ba 2 days ago 968MB
<none> <none> e4ee648c5b12 3 days ago 975MB
redis latest 29ab4501eac3 3 days ago 117MB
kicbase/stable v0.0.36 866c1fe4e3f2 8 weeks ago 1.11GB
docker <none> 0dfb722b0a54 3 months ago 134MB

heliahemipour@MacBook-Pro ~% cd /Users/heliahashemipour/Desktop/Cloud/HW2-0931106/step1 % docker rmi heliahemipour/new-ubuntu:1.0
Untagged: heliahemipour/new-ubuntu:1.0
Untagged: heliahemipour/new-ubuntu@sha256:e77e56a2bed143af33c30866d76d9fc3a54287a0e4d0d8ee71845a637afa9098c
Deleted: sha256:36045a49bf0e66f0256df7132d659de0804bc6232f4bd9f46fed5f

heliahemipour@MacBook-Pro ~% cd /Users/heliahashemipour/Desktop/Cloud/HW2-0931106/step1 % docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
heliahemipour/coin-server 1.0 3c6ab8c04975 4 hours ago 71.8MB
coin-server 1.0 3c6ab8c04975 4 hours ago 71.8MB
heliahemipour/coin-app <none> f7bfa443d9 12 hours ago 78.3MB
tiangolo/uwsgi-nginx-flask latest 9634df2032ba 2 days ago 968MB
<none> <none> e4ee648c5b12 4 days ago 975MB
redis latest 29ab4501eac3 4 days ago 117MB
heliahemipour/new-alpine 1.0 48cf7a587854 5 days ago 11.9MB
kicbase/stable v0.0.36 866c1fe4e3f2 8 weeks ago 1.11GB
docker <none> 0dfb722b0a54 3 months ago 134MB

heliahemipour@MacBook-Pro ~% cd /Users/heliahashemipour/Desktop/Cloud/HW2-0931106/step1 % docker pull heliahemipour/new-ubuntu:1.0
1.0: Pulling from heliahemipour/new-ubuntu
846c0181ff: Already exists
3bb20723e9c4: Already exists
Digest: sha256:e7e56a2bed143af33c30866d76d9fc3a54287a0e4d0d8ee71845a637afa9098c
Status: Downloaded newer image for heliahemipour/new-ubuntu:1.0
docker.io/heliahashemipour/new-ubuntu:1.0

heliahemipour@MacBook-Pro ~% cd /Users/heliahashemipour/Desktop/Cloud/HW2-0931106/step1 % docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
heliahemipour/coin-server 1.0 3c6ab8c04975 4 hours ago 71.8MB
coin-server 1.0 3c6ab8c04975 4 hours ago 71.8MB
heliahemipour/coin-app <none> f7bfa443d9 12 hours ago 78.3MB
heliahemipour/new-ubuntu 1.0 036b4b5b498f 12 hours ago 129MB
tiangolo/uwsgi-nginx-flask latest 9634df2032ba 2 days ago 968MB
<none> <none> e4ee648c5b12 4 days ago 975MB
redis latest 29ab4501eac3 4 days ago 117MB
kicbase/stable v0.0.36 866c1fe4e3f2 8 weeks ago 1.11GB
heliahemipour/new-alpine 1.0 48cf7a587854 5 days ago 11.9MB
The document has moved
<A HREF="http://www.google.com/">here</A>.
</BODY></HTML>
root@lbfcafadb7cb:/# curl google.com
<HTML><HEAD><meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<TITLE>301 Moved</TITLE><HEAD><BODY>
<H1>301 Moved</H1>

```

گام دوم

• دریافت ایمیج ردیس و ساختن کانتینر از آن، ساختن شبکه و ساختن Volume

ابتدا ایمیج ردیس را pull کرده و آن را اجرا میکنیم از اطلاعات کش ردیس را persist میکنیم بنابراین از قابلیت volume استفاده میکنیم. همچنین برای برقراری ارتباط بین دو کانتینر بسازیم.

```
heliaa@MacBook-Pro ~ % docker pull redis
Using default tag: latest
latest: Pulling from library/redis
Digest: sha256:fdcaa0102e0c66802845aa5c961cb89a091a188056811802383660cd9e10889da
Status: Image is up to date for redis:latest
docker.io/library/redis:latest

heliaa@MacBook-Pro ~ % docker network create flask
b5b51ce76c8fc661e58e7c715faee4960605343f5c855b73723c2f1d23fe9a20

heliaa@MacBook-Pro ~/University/Semester7/Cloud/IIR-933106/step2 % docker run --rm --network flask -v ./data --name redis redis redis-server --appendonly yes
1:C 21 Dec 2022 11:36:32.048 # O0000000000 Redis is starting o000000000000
1:C 21 Dec 2022 11:36:32.048 # Redis version=7.0.7, bits=64, commit=00000000, modified=0, pid=1, just started
1:C 21 Dec 2022 11:36:32.048 # Configuration loaded
1:M 21 Dec 2022 11:36:32.049 * monotonic clock: POSIX clock_gettime
1:M 21 Dec 2022 11:36:32.054 * Running mode=standalone, port=6379.
1:M 21 Dec 2022 11:36:32.054 # Server initialized
1:M 21 Dec 2022 11:36:32.063 * Creating AOF base file appendonly.aof.1.base.rdb on server start
1:M 21 Dec 2022 11:36:32.073 * Creating AOF incr file appendonly.aof.1.incr.aof on server start
1:M 21 Dec 2022 11:36:32.073 * Ready to accept connections
```

• ساختن ایمیج سرور نوشته شده با استفاده از داکرفایل

حال یک داکرفایل مینویسیم با توجه به فرض خواسته شده environment variable (که شامل شماره پورتی که بر روی آن سرور خود را بالا میاورید، مدت زمان اعتبار کلیدها در کش ردیس و نام رمز ارزی که قیمت آن را میخواهیم) در آن ست میکنیم. از طرفی کدهای مربوطه و requirementها را نیز در کانتینر کپی میکنیم و requirementهای مربوطه در فایل requirement.txt را نصب میکنیم. سپس آن را build میکنیم.

```
FROM python:3.11.0a6-alpine3.15

LABEL maintainer="Helia Hashemipour <heliahashemipour2@gmail.com>

WORKDIR /app

COPY requirements.txt .

RUN pip install -r requirements.txt

COPY . .

CMD python app.py
```

```
heliaa@MacBook-Pro ~ /University/Semester7/Cloud/HW2-9831106/step2 ➜ main ➔ docker build -t coin-server:1.0 .
[+] Building 3.0s (11/11) FINISHED
=> [internal] load build definition from Dockerfile          0.0s
=> => transferring dockerfile: 127B                         0.0s
=> [internal] load .dockerignore                           0.0s
=> => transferring context: 2B                            0.0s
=> [internal] load metadata for docker.io/library/python:3.11.0a6-alpine 2.9s
=> [auth] library/python:pull token for registry-1.docker.io
=> [1/5] FROM docker.io/library/python:3.11.0a6-alpine3.15@sha256:6c215d 0.0s
=> [internal] load build context                          0.0s
=> => transferring context: 230B                         0.0s
=> CACHED [2/5] WORKDIR /app                            0.0s
=> CACHED [3/5] COPY requirements.txt .                  0.0s
=> CACHED [4/5] RUN pip install -r requirements.txt    0.0s
=> CACHED [5/5] COPY . .                                0.0s
=> exporting to image                                    0.0s
=> => exporting layers                                  0.0s
=> => writing image sha256:3ca6b8c04975759679cc79fb68ac70ea4641b59ec6b84 0.0s
=> => naming to docker.io/library/coin-server:1.0        0.0s
```

در این مرحله ایمیج را ران میکنیم.

```
heliaa@MacBook-Pro ~ /University/Semester7/Cloud/HW2-9831106/step2 ➜ main ➔ docker run -it --rm -e SERVER_PORT="80" -e COIN_NAME="bitcoin" -e CACHE_EXPIRE_TIME="300" -p 80:80 --network flask --name coin-price coin-server:1.0
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:80
 * Running on http://172.20.0.3:80
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 823-639-635
172.20.0.1 - - [21/Dec/2022 10:23:44] "GET / HTTP/1.1" 200 -
```

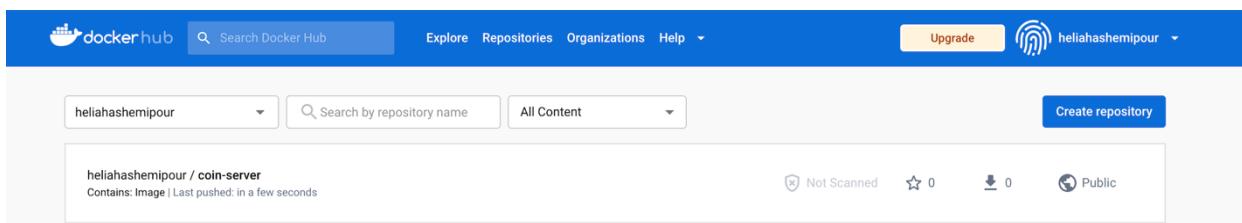
حال با دستور کرل خروجی را می‌گیریم.

```
curl localhost
```

{
 "name": "bitcoin",
 "price": "16869.18"
}

• ارسال ایمیج ساخته شده بر روی داکرهاب و نمایش نتیجه آن

```
heliaa@MacBook-Pro ~ /University/Semester7/Cloud/HW2-9831106/step2 ➜ main ➔ docker tag coin-server:1.0 heliahashemipour/coin-server:1.0
heliaa@MacBook-Pro ~ /University/Semester7/Cloud/HW2-9831106/step2 ➜ main ➔ docker image push heliahashemipour/coin-server:1.0
The push refers to repository [docker.io/heliahashemipour/coin-server]
c7abf0c13001: Mounted from heliahashemipour/coin-app
f5d9755716b8: Pushed
40c3830087a4: Pushed
0fid7ebafibe: Pushed
cf47a6e12eb3: Mounted from library/python
e643e0a6e20e: Pushed
2091c8163eb1: Mounted from library/python
fb07d5451c69: Mounted from library/python
4fc242d58285: Mounted from library/python
```



نمایش اطلاعات ایمیج سرور خود با استفاده از دستور docker inspect

```
heliae@MacBook-Pro ~ % docker image inspect coin-server:1.0
[{"Id": "sha256:3ca6b8c04975759679cc79fb68ac70ea4641b59ec6b849e6f4f92d1504797488",
 "RepoTags": [
   "heliahemipour/coin-server:1.0",
   "coin-server:1.0"
 ],
 "RepoDigests": [
   "heliahemipour/coin-server@sha256:e060707a41b194f6846c665d9eade9b794aff197f2d384a6b5cfec010e23f8e4"
 ],
 "Parent": "",
 "Comment": "buildkit.dockerfile.v0",
 "Created": "2022-12-21T09:30:48.438751294Z",
 "Container": "",
 "ContainerConfig": {
   "Hostname": "",
   "Domainname": "",
   "User": "",
   "AttachStdin": false,
   "AttachStdout": false,
   "AttachStderr": false,
   "Tty": false,
   "OpenStdin": false,
   "StdinOnce": false,
   "Env": null,
   "Cmd": null,
   "Image": "",
   "Volumes": null,
   "WorkingDir": "",
   "Entrypoint": null,
   "OnBuild": null,
   "Labels": null
 },
 "DockerVersion": "",
 "Author": "",
 "Config": {
   "Hostname": "",
   "Domainname": "",
   "User": "",
   "AttachStdin": false,
   "AttachStdout": false,
   "AttachStderr": false,
   "Tty": false,
   "OpenStdin": false,
   "StdinOnce": false,
   "Env": [
     "PATH=/usr/local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin",
     "LANG=C.UTF-8"
   ]
 }
```

```
heliae@MacBook-Pro ~ % docker image inspect redis
[{"Id": "sha256:29ab4501eac3b347f1251ef5b9eff1a79b71e6736706aca43ca98a7e9cd9dcf7",
 "RepoTags": [
   "redis:latest"
 ],
 "RepoDigests": [
   "redis@sha256:fdaa0102e0c66802845aa5c961cb89a091a188056811802383660cd9e10889da"
 ],
 "Parent": "",
 "Comment": "",
 "Created": "2022-12-17T01:58:56.198976747Z",
 "Container": "5e0c984f92c64338cb0e634dde9d06b3af8d14236ba19c36bf74011c22d0d153d",
 "ContainerConfig": {
   "Hostname": "5e0c984f92c6",
   "Domainname": "",
   "User": "",
   "AttachStdin": false,
   "AttachStdout": false,
   "AttachStderr": false,
   "ExposedPorts": {
     "6379/tcp": {}
   },
   "Tty": false,
   "OpenStdin": false,
   "StdinOnce": false,
   "Env": [
     "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin",
     "GOSU_VERSION=1.14",
     "REDIS_VERSION=7.0.7",
     "REDIS_DOWNLOAD_URL=http://download.redis.io/releases/redis-7.0.7.tar.gz",
     "REDIS_DOWNLOAD_SHA=8d327d7e887d1bb308fc37aa71a0bf79f58129e3739069aaeeae88955ac586"
   ],
   "Cmd": [
     "/bin/sh",
     "-c",
     "#(nop)",
     "CMD [\"redis-server\"]"
   ],
   "Image": "sha256:3a96b43e7d4690f4162b792ba9ec63ffbb3abfied29639746ae964113fd115a04",
   "Volumes": {
     "/data": {}
   },
   "WorkingDir": "/data",
   "Entrypoint": [
     "docker-entrypoint.sh"
   ],
   "OnBuild": null
 }
```

• نمایش کانتینرهای موجود در سیستم خود با استفاده از دستور docker ps

```
x heliaa@MacBook-Pro ~ /University/Semester7/Cloud/HW2-9831106/step2 ] docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
 NAMES
0a5c8d1103dc redis "docker-entrypoint.s..." 4 minutes ago Up 4 minutes 6379/tcp
redis
61a4b390ca13 coin-server:1.0 "/bin/sh -c 'python ..." 4 minutes ago Up 4 minutes 0.0.0.0:80->80/tcp
coin-price
a865b55d33cc kicbase/stable:v0.0.36 "/usr/local/bin/entr..." 24 hours ago Up About an hour 0.0.0.0:54885->22/tcp, 0.0.0.0:54886->2376/tcp, 0.0.0.0:54888->5000/tcp, 0.0.0.0:54889->8443/tcp, 0.0.0.0:54887->32443/tcp minikube
```

• نمایش میزان منابع استفاده شده توسط کانتینر های موجود با استفاده از دستور docker stats

CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %	NET I/O	BLOCK I/O	PIDS
0a5c8d1103dc	redis	0.36%	2.66MiB / 3.842GiB	0.07%	1.88kB / 686B	0B / 16.4kB	6
61a4b390ca13	coin-price	0.25%	56.34MiB / 3.842GiB	1.43%	7.72kB / 3kB	0B / 291kB	3
a865b55d33cc	minikube	25.51%	906.3MiB / 2.148GiB	41.20%	95.7MB / 3.02MB	111MB / 349MB	548

گام سوم

برای این گام ابتدا minikube start را میزنیم. سپس فایل های مورد نیاز پروژه را مینویسیم. از Redis شروع میکنیم. ابتدا فایل آن به صورت زیر میباشد.

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: redis
spec:
  storageClassName: manual
  accessModes:
    - ReadWriteOnce
  capacity:
    storage: 100Mi
  hostPath:
    path: /data/redis/
```

حال آن را با دستور زیر میکنیم.

```
heliaa@MacBook-Pro ~ /University/Semester7/Cloud/HW2-9831106/step3 ] docker exec -it minikube bash
[root@minikube ~]# kubectl apply -f redis-pv.yaml
persistentvolume/redis created
```

به همین منوال فایل های redis-service.yaml ، redis-deployment.yaml ، redis-pvc.yaml را مینویسیم (فایلها ضمیمه شده اند) و به ترتیب ذکر شده اپلای میکنیم.

```
heliaa@MacBook-Pro ~ /University/Semester7/Cloud/HW2-9831106/step3 ] docker exec -it minikube bash
[root@minikube ~]# kubectl apply -f redis-pvc.yaml
persistentvolumeclaim/redis-pvc created
[root@minikube ~]# kubectl apply -f redis-deployment.yaml
deployment.apps/redis created
[root@minikube ~]# kubectl apply -f redis-service.yaml
```

حال سراغ فایل های سرور خود میرویم. ابتدا فایل یک سری تنظیمات برنامه مانند آدرسی که برنامه باید به آن درخواست ارسال کند، پورت سرور، نام رمز ارز و مدت زمان نگهداری کش و مشخص میشود. (دستور

Kubectl create cm -n <namespace> <name> --dry-run=client -o yaml>cm.yaml برای ساختن فایل است)

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: server-config
data:
  SERVER_PORT: "80"
  HOST_API: "https://api.coingecko.com/api/v3/simple/vs_currencies=usd&price?ids="
  COIN_NAME: "bitcoin"
  CACHE_EXPIRE_TIME: "300"
```

سپس فایل deployment را با دستور زیر میسازیم.(تعداد replicas هم ۲ است)

```
| heliaa@MacBook-Pro ~$ cd University/Semester7/Cloud/HW2-9831106/step3 | main $ kubectl create deployment server-deployment --replicas=2 --image=heliaahashemipour/coin-server:1.0 --dry-run=client -o yaml |> server-deployment.yaml
```

- در فایل ساخته شده و در env، مقادیر را از configmap میخوانیم. سپس آن را نیز اپلای میکنیم.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: server-deployment
  labels:
    app: server-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: server-deployment
  template:
    metadata:
      labels:
        app: server-deployment
    spec:
      containers:
        - name: coin-server
          image: heliaahashemipour/coin-server:1.0
          ports:
            - containerPort: 80
          env:
            - name: SERVER_PORT
              valueFrom:
                configMapKeyRef:
                  name: server-config
                  key: SERVER_PORT
            - name: COIN_NAME
              valueFrom:
                configMapKeyRef:
                  name: server-config
                  key: COIN_NAME
            - name: HOST_API
              valueFrom:
                configMapKeyRef:
                  name: server-config
                  key: HOST_API
            - name: CACHE_EXPIRE_TIME
              valueFrom:
                configMapKeyRef:
                  name: server-config
                  key: CACHE_EXPIRE_TIME
```

فایل server-service.yaml هم به صورت زیر است.

```
apiVersion: v1
kind: Service
metadata:
  name: server-service
spec:
  selector:
    app: server-deployment
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
```

حال به ترتیب توضیح داده شده apply میکنیم.

```
| heliaa@MacBook-Pro ~$ cd University/Semester7/Cloud/HW2-9831106/step3 | main $ kubectl apply -f server-config.yaml
configmap/server-config created
| x heliaa@MacBook-Pro ~$ cd University/Semester7/Cloud/HW2-9831106/step3 | main $ kubectl apply -f server-deployment.yaml
deployment.apps/server-deployment created
| heliaa@MacBook-Pro ~$ cd University/Semester7/Cloud/HW2-9831106/step3 | main $ kubectl apply -f server-service.yaml
service/server-service created
```

- با استفاده از دستور `kubectl get` صحت ایجاد منابع (کانفیگمپ، دیپلومینت، سرویس و پادها، PVC) را بر روی کلاستر Minikube نشان دادیم.

```
heliaa@MacBook-Pro:~/University/Semester7/Cloud/HW2-9831106/step4$ main $ kubectl get configmap
NAME          DATA   AGE
kube-root-ca.crt 1      23h
server-config  4      63m

heliaa@MacBook-Pro:~/University/Semester7/Cloud/HW2-9831106/step4$ main $ kubectl get deployment
NAME        READY  UP-TO-DATE AVAILABLE AGE
redis       1/1    1           1      96m
server-deployment 2/2    2           2      63m

heliaa@MacBook-Pro:~/University/Semester7/Cloud/HW2-9831106/step4$ main $ kubectl get service
NAME        TYPE     CLUSTER-IP      EXTERNAL-IP   PORT(S)   AGE
Kubernetes  ClusterIP   10.96.0.1    <none>        443/TCP   23h
redis       ClusterIP   10.98.159.148 <none>        6379/TCP  96m
server-service ClusterIP   10.99.19.217 <none>        80/TCP   63m

[ heliaa@MacBook-Pro:~/University/Semester7/Cloud/HW2-9831106/step4$ main $ kubectl get pv
NAME    CAPACITY ACCESS MODES RECLAIM POLICY STATUS CLAIM STORAGECLASS REASON AGE
redis  100Mi   RWO      Retain   Bound   default/redis-pvc manual   96m

heliaa@MacBook-Pro:~/University/Semester7/Cloud/HW2-9831106/step4$ main $ kubectl get pvc
NAME    STATUS VOLUME CAPACITY ACCESS MODES STORAGECLASS AGE
redis  Bound   redis  100Mi   RWO      manual   96m

heliaa@MacBook-Pro:~/University/Semester7/Cloud/HW2-9831106/step4$ main $ kubectl get pods -o wide
NAME        READY  STATUS RESTARTS AGE   IP          NODE NOMINATED NODE READINESS GATES
redis-94d549f8-xdpch 1/1   Running 0 97m  172.17.0.5  minikube  <none>   <none>
server-deployment-59b8cf9fd-vsxz6 1/1   Running 0 64m  172.17.0.6  minikube  <none>   <none>
server-deployment-59b8cf9fd-zm5tb 1/1   Running 0 64m  172.17.0.7  minikube  <none>   <none>
```

- آدرس IP پادها را با استفاده از endpoint

```
[ heliaa@MacBook-Pro:~/University/Semester7/Cloud/HW2-9831106/step4$ main $ kubectl get ep
NAME        ENDPOINTS   AGE
kubernetes  192.168.49.2:8443 23h
redis       172.17.0.5:6379  93m
server-service 172.17.0.6:80,172.17.0.7:80 60m
```

گام چهارم

ابتدا با ایمیج ubuntu تست را انجام میدهیم.

```
heliaa@MacBook-Pro ~/University/Semester7/Cloud/HW2-9831100/step4 % main $ kubectl run my-shell --rm -i --tty --image heliahashemipour/new-ubuntu1.0 -- bash
If you don't see a command prompt, try pressing enter.
root@my-shell:/# curl server-service:80
{
  "name": "bitcoin",
  "price": "16844.53"
}
root@my-shell:/# curl server-service:80
{
  "name": "bitcoin",
  "price": "16844.53"
}
root@my-shell:/# curl server-service:80
{
  "name": "bitcoin",
  "price": "16844.53"
}
root@my-shell:/# curl server-service:80
{
  "name": "bitcoin",
  "price": "16844.53"
}
root@my-shell:/# curl server-service:80
{
  "name": "bitcoin",
  "price": "16844.53"
}
root@my-shell:/# curl server-service:80
{
  "name": "bitcoin",
  "price": "16844.53"
}
root@my-shell:/# curl server-service:80
{
  "name": "bitcoin",
  "price": "16844.53"
}
root@my-shell:/# curl server-service:80
{
  "name": "bitcoin",
  "price": "16844.53"
}
root@my-shell:/# curl server-service:80
{
  "name": "bitcoin",
  "price": "16844.53"
}
root@my-shell:/# curl server-service:80
{
  "name": "bitcoin",
  "price": "16844.53"
}
root@my-shell:/# curl server-service:80
{
  "name": "bitcoin",
  "price": "16844.53"
}
root@my-shell:/# curl server-service:80
{
  "name": "bitcoin",
  "price": "16844.53"
}
```

چند دستور curl را میزنیم که تمامی دستورات با موفقیت نتیجه داده‌اند (البته جز یکی که به علت کندی اینترنت مجبور شدم بزن) و با گرفتن log از دو تا پاد بار یکسان انجام شده توزیع شده است.

```
heliaa@MacBook-Pro ~/University/Semester7/Cloud/HW2-9831100/step4 % main $ kubectl logs server-deployment-59b8c9f59d-zm5tb
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:80
* Running on https://172.17.0.7:80
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 975-2421-500
172.17.0.1 -- [21/Dec/2022 12:16:00] "GET / HTTP/1.1" 200 -
172.17.0.1 -- [21/Dec/2022 12:18:55] "GET / HTTP/1.1" 200 -
172.17.0.1 -- [21/Dec/2022 12:20:23] "GET / HTTP/1.1" 500
172.17.0.1 -- [21/Dec/2022 12:20:40] "GET / HTTP/1.1" 200
172.17.0.1 -- [21/Dec/2022 12:41:42] "GET / HTTP/1.1" 200
172.17.0.1 -- [21/Dec/2022 12:41:43] "GET / HTTP/1.1" 200
172.17.0.1 -- [21/Dec/2022 12:41:44] "GET / HTTP/1.1" 200
172.17.0.1 -- [21/Dec/2022 12:41:47] "GET / HTTP/1.1" 200
172.17.0.1 -- [21/Dec/2022 12:41:48] "GET / HTTP/1.1" 200
172.17.0.1 -- [21/Dec/2022 12:41:50] "GET / HTTP/1.1" 200 -
172.17.0.1 -- [21/Dec/2022 12:41:52] "GET / HTTP/1.1" 200 -
heliaa@MacBook-Pro ~/University/Semester7/Cloud/HW2-9831100/step4 % main $ kubectl logs server-deployment-59b8c9f59d-vsxz6
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:80
* Running on https://172.17.0.6:80
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 238-184-845
172.17.0.1 -- [21/Dec/2022 11:12:30] "GET / HTTP/1.1" 200 -
172.17.0.1 -- [21/Dec/2022 12:20:39] "GET / HTTP/1.1" 200 -
172.17.0.1 -- [21/Dec/2022 12:20:41] "GET / HTTP/1.1" 200 -
172.17.0.1 -- [21/Dec/2022 12:41:45] "GET / HTTP/1.1" 200 -
172.17.0.1 -- [21/Dec/2022 12:41:46] "GET / HTTP/1.1" 200 -
172.17.0.1 -- [21/Dec/2022 12:41:46] "GET / HTTP/1.1" 200 -
172.17.0.1 -- [21/Dec/2022 12:41:47] "GET / HTTP/1.1" 200 -
172.17.0.1 -- [21/Dec/2022 12:41:48] "GET / HTTP/1.1" 200 -
172.17.0.1 -- [21/Dec/2022 12:41:49] "GET / HTTP/1.1" 200 -
172.17.0.1 -- [21/Dec/2022 12:41:50] "GET / HTTP/1.1" 200 -
172.17.0.1 -- [21/Dec/2022 12:41:51] "GET / HTTP/1.1" 200 -
```

حال با ایمیج alpine امتحان میکنیم. ابتدا با دستور `kubectl run` سعی میکنیم ایمیج ساخته شده در را از آن یک pod درست کنیم.

```
[heliaa@MacBook-Pro ~]$ kubectl run curl-pods --image=heliahashemipour/new-alpine:1.0  
pod/curl-pods created  
[heliaa@MacBook-Pro ~]$ kubectl get pods  
NAME          READY   STATUS    RESTARTS   AGE  
curl-pods     0/1     CrashLoopBackOff  1 (2s ago)  4s  
redis-94d5459f8-xdpch  1/1     Running   0          141m  
server-deployment-59b8c9f59d-vsx26  1/1     Running   0          108m  
server-deployment-59b8c9f59d-zmtb  1/1     Running   0          108m  
[heliaa@MacBook-Pro ~]$ kubectl delete pod curl-pods  
pod "curl-pods" deleted
```

مشاهده میکنیم status آن به CrashLoopBackOff رسیده است. یک deployment ساخته که در آن هنگام اجرای container دستور bin/sleep/ را به مدت بینهایت اجرا کند. با این کار دیگر کارش به پایان نرسیده و میتوانیم دسترسی به ash را بگیریم. بنابراین در ابتدا pod ساخته شده را حذف میکنیم.

```

heliaa@MacBook-Pro ~% cd /University/Semester7/Cloud/HW2-9831100/step4
$ kubectl create deployment curl-deployment --image=heliaahashemipour/new-alpine:1.0 --dry-run=client -o yaml > curl-deployment.yaml
$ kubectl apply -f curl-deployment.yaml
deployment.apps/curl-deployment created
$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
curl-deployment-5766697bf-4n25k   1/1     Running   0          11s
redis-94d5459f8-xdpch   1/1     Running   0          143m
server-deployment-598b9f59d-vsxz6   1/1     Running   0          11m
server-deployment-598b9f59d-zmtbz   1/1     Running   0          11m

```

به پاد مورد نظر وصل شده و دسترسی ash را می‌گیریم.

```
liaoMacBook-Pro: ~ % kubectl logs server-deployment-598c9f59d-vsxdz [main] kubectl logs server-deployment-598c9f59d-vsxdz
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:80
* Running on https://127.0.0.1:80
Press CTRL+C to quit
* Restarting with stat
* Debugger is active
* Debugger PIN: 238-184-045
172.17.0.1 -- [21/Oct/2022 11:12:39] "GET / HTTP/1.1" 200 -
172.17.0.1 -- [21/Oct/2022 12:20:39] "GET / HTTP/1.1" 200 -
172.17.0.1 -- [21/Oct/2022 12:20:41] "GET / HTTP/1.1" 200 -
172.17.0.1 -- [21/Oct/2022 12:41:45] "GET / HTTP/1.1" 200 -
172.17.0.1 -- [21/Oct/2022 12:41:46] "GET / HTTP/1.1" 200 -
172.17.0.1 -- [21/Oct/2022 12:41:46] "GET / HTTP/1.1" 200 -
172.17.0.1 -- [21/Oct/2022 12:41:47] "GET / HTTP/1.1" 200 -
172.17.0.1 -- [21/Oct/2022 12:41:48] "GET / HTTP/1.1" 200 -
172.17.0.1 -- [21/Oct/2022 12:41:49] "GET / HTTP/1.1" 200 -
172.17.0.1 -- [21/Oct/2022 12:41:58] "GET / HTTP/1.1" 200 -
172.17.0.1 -- [21/Oct/2022 12:41:51] "GET / HTTP/1.1" 200 -
172.17.0.1 -- [21/Oct/2022 12:59:52] "GET / HTTP/1.1" 200 -
172.17.0.1 -- [21/Oct/2022 12:59:53] "GET / HTTP/1.1" 200 -
172.17.0.1 -- [21/Oct/2022 12:59:54] "GET / HTTP/1.1" 200 -
172.17.0.1 -- [21/Oct/2022 12:59:58] "GET / HTTP/1.1" 200 -
172.17.0.1 -- [21/Oct/2022 13:00:00] "GET / HTTP/1.1" 200 -
liaoMacBook-Pro: ~ % kubectl logs server-deployment-598c9f59d-zsdtb [main] kubectl logs server-deployment-598c9f59d-zsdtb
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:80
* Running on https://127.17.0.7:80
Press CTRL+C to quit
* Restarting with stat
* Debugger is active
* Debugger PIN: 975-241-680
172.17.0.1 -- [21/Oct/2022 12:16:00] "GET / HTTP/1.1" 200 -
172.17.0.1 -- [21/Oct/2022 12:18:55] "GET / HTTP/1.1" 200 -
172.17.0.1 -- [21/Oct/2022 12:20:23] "GET / HTTP/1.1" 500 -
172.17.0.1 -- [21/Oct/2022 12:20:40] "GET / HTTP/1.1" 200 -
172.17.0.1 -- [21/Oct/2022 12:41:42] "GET / HTTP/1.1" 200 -
172.17.0.1 -- [21/Oct/2022 12:41:43] "GET / HTTP/1.1" 200 -
172.17.0.1 -- [21/Oct/2022 12:41:44] "GET / HTTP/1.1" 200 -
172.17.0.1 -- [21/Oct/2022 12:41:47] "GET / HTTP/1.1" 200 -
172.17.0.1 -- [21/Oct/2022 12:41:48] "GET / HTTP/1.1" 200 -
172.17.0.1 -- [21/Oct/2022 12:41:50] "GET / HTTP/1.1" 200 -
172.17.0.1 -- [21/Oct/2022 12:41:52] "GET / HTTP/1.1" 200 -
172.17.0.1 -- [21/Oct/2022 12:59:49] "GET / HTTP/1.1" 200 -
172.17.0.1 -- [21/Oct/2022 12:59:51] "GET / HTTP/1.1" 200 -
172.17.0.1 -- [21/Oct/2022 12:59:55] "GET / HTTP/1.1" 200 -
172.17.0.1 -- [21/Oct/2022 12:59:56] "GET / HTTP/1.1" 200 -
172.17.0.1 -- [21/Oct/2022 13:00:01] "GET / HTTP/1.1" 200 -
```

باز هم با گرفتن \log از دو تا پاد میبینیم که بین ۲ پاد بار یکسان انجام شده توزیع شده است.

دستور get و مشاهده تمامی منابع

```

helia@MacBook-Pro ~]$ kubectl get all
NAME           READY   STATUS    RESTARTS   AGE
pod/curl-deployment-5766697b7f-4n2k   1/1    Running   0          49m
pod/redis-94d5459f8-xdpch   1/1    Running   0          3h13m
pod/server-deployment-59b8c9f59d-vs2x6  1/1    Running   0          160m
pod/server-deployment-59b8c9f59d-zn5tb  1/1    Running   0          160m

NAME            TYPE        CLUSTER-IP      EXTERNAL-IP     PORT(S)   AGE
service/kubernetes   ClusterIP   10.96.0.1       <none>        443/TCP   25h
service/redis       ClusterIP   10.98.159.140   <none>        6379/TCP   3h13m
service/server-service ClusterIP  10.99.19.217   <none>        88/TCP    160m

NAME           READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/curl-deployment  1/1      1           1          49m
deployment.apps/redis             1/1      1           1          3h13m
deployment.apps/server-deployment 2/2      2           2          160m

NAME           DESIRED   CURRENT   READY   AGE
replicaset.apps/curl-deployment-5766697b7f  1         1         1         49m
replicaset.apps/redis-94d5459f8                1         1         1         3h13m
replicaset.apps/server-deployment-59b8c9f59d  2         2         2         160m

```