

گزارش نهایی پروژه ریانش ابری

محمدجواد زندیه ۹۸۳۱۰۳۲

امیرمحمد بابایی ۹۸۳۱۰۱۱

۱. ساخت ایمیج با استفاده از Dockerfile نوشته شده:

با استفاده از دستور `-d docker-compose up` ایمیج های مورد ساخته میشوند و کانتینرهای آنها نیز بالا می آیند:

```
./run.sh
[+] Running 2/2
✓ Network aut-cc-final-project_default  Created
✓ Container mysql_db                  Started
[+] Running 1/1
✓ Container coin_news                Started
[+] Running 3/3
✓ Container coin_news                Running
✓ Container mysql_db                 Running
✓ Container bepa                     Started
[+] Running 2/2
✓ Container mysql_db                 Running
✓ Container peyk                     Started
[+] Running 1/1
✓ Container react                   Started
```

docker images				
REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
javad0zandiye/aut-cc-final-react	v1	07c97a26aa92	4 hours ago	481MB
react	v1	07c97a26aa92	4 hours ago	481MB
javad0zandiye/aut-cc-final-peyk	v1	ba099d60ea83	44 hours ago	98.3MB
peyk	v1	ba099d60ea83	44 hours ago	98.3MB
javad0zandiye/aut-cc-final-bepa	v1	9615af85359	44 hours ago	65.6MB
bepa	v1	9615af85359	44 hours ago	65.6MB
node	lts-alpine3.18	ab464216c5ef	7 days ago	173MB
python	3.9.17-alpine	999a145e6854	3 weeks ago	53.1MB
mysql	8.0	b3077e10c878	3 weeks ago	592MB
amirhossein21/coinnews.macos	v0.2	d8cdc24b78a0	4 weeks ago	484MB

docker ps -a				
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
27b69688df3	react:v1	"/bin/sh entrypoint..."	37 seconds ago	Up 35 seconds
96b5d3fbf3b4	peyk:v1	"/bin/sh entrypoint..."	38 seconds ago	Up 37 seconds
3ba079afddb4	bepa:v1	"python bepa.py"	39 seconds ago	Up 38 seconds
5beeb3506ca0	amirhossein21/coinnews.macos:v0.2	"/bin/sh -c './main ..."	40 seconds ago	Up 39 seconds
f35d0670bfc2	mysql:8.0	"docker-entrypoint.s..."	41 seconds ago	Up 39 seconds

docker-compose ls				
NAME	STATUS	CONFIG FILES		
aut-cc-final-project	running(5)	/Users/javad/Desktop/files/university-courses/cloud-computing/home-works/final-project/AUT-CC-final-project/docker-compose.yml		

aut-cc-final-project		Running (5/5)	2 minutes ago	⋮	⋮
mysql_db	f35d0670bfc2	mysql:8.0	Running	3306:3306	2 minutes ago
coin_news	5beeb3506ca0	amirhossein21/coinnews.macos:v0.2	Running	8000:8000	2 minutes ago
bepa	3ba079afddb4	bepa:v1	Running	8002:8002	2 minutes ago
peyk	96b5d3fbf3b4	peyk:v1	Running	8001:8001	2 minutes ago
react	27b69688df3	react:v1	Running	3000:3000	2 minutes ago

<code>javad0zandiye/h/aut-cc-final-react</code>	v1	In use	about 4 hours	480.96 MB	▶	⋮	☰
<code>react</code>	v1	In use	about 4 hours	480.96 MB	▶	⋮	☰
<code>javad0zandiye/h/aut-cc-final-peyk</code>	v1	In use	2 days ago	98.33 MB	▶	⋮	☰
<code>peyk</code>	v1	In use	2 days ago	98.33 MB	▶	⋮	☰
<code>javad0zandiye/h/aut-cc-final-bepa</code>	v1	In use	2 days ago	65.57 MB	▶	⋮	☰
<code>bepa</code>	v1	In use	2 days ago	65.57 MB	▶	⋮	☰
<code>node</code>	lts-alpine3.18	Unused	8 days ago	172.75 MB	▶	⋮	☰
<code>python</code>	3.9.17-alpine	Unused	23 days ago	53.13 MB	▶	⋮	☰
<code>mysql</code>	8.0	In use	27 days ago	592.43 MB	▶	⋮	☰

برای ساخت ایمیج ها از فایل docker-compose استفاده شده است که جزو موارد امتیاز پژوه است.

۲. ارسال ایمیج ساخته شده بر روی داکرهاب و نتیجه آن

ابتدا با استفاده از دستور docker tag به ایمیج ها تگ مناسب برای ارسال روش داکرهاب را میدهیم و سپس با دستور

docker push ایمیج را روی داکرهاب قرار میدهیم:

 [javad0zandiye/h / aut-cc-final-react](#)

Description
This repository does not have a description 

 Last pushed: an hour ago

Tags
This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
v1		Image	---	an hour ago

[See all](#) [Go to Advanced Image Management](#)

Docker commands [Public View](#)

To push a new tag to this repository,

```
docker push javad0zandiye/h/aut-cc-final-react:tagname
```

Automated Builds

Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Available with Pro, Team and Business subscriptions. [Read more about automated builds ↗](#)

[Upgrade](#)

 [javad0zandiye/h / aut-cc-final-peyk](#)

Description
This repository does not have a description 

 Last pushed: an hour ago

Tags
This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
v1		Image	---	an hour ago

[See all](#) [Go to Advanced Image Management](#)

Docker commands [Public View](#)

To push a new tag to this repository,

```
docker push javad0zandiye/h/aut-cc-final-peyk:tagname
```

Automated Builds

Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Available with Pro, Team and Business subscriptions. [Read more about automated builds ↗](#)

[Upgrade](#)

javad0zandiye / aut-cc-final-bepa

Description
This repository does not have a description [Edit](#)

Last pushed: 2 hours ago

Tags
This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
v1	🐧	Image	...	2 hours ago

[See all](#) [Go to Advanced Image Management](#)

Docker commands
To push a new tag to this repository,
`docker push javad0zandiye / aut-cc-final-bepa:tagname`

[Public View](#)

Automated Builds
Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.
Available with Pro, Team and Business subscriptions. [Read more about automated builds](#).

[Upgrade](#)

javad0zandiye / aut-cc-final-react
Contains: Image | Last pushed: an hour ago

javad0zandiye / aut-cc-final-peyk
Contains: Image | Last pushed: an hour ago

javad0zandiye / aut-cc-final-bepa
Contains: Image | Last pushed: an hour ago

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 اجرای نیست ولی توصیه میشود)

POST | <http://127.0.0.1:8001/subscribe/> [Send](#)

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```

1 {
2   "email": "amirmohamad.babaei79@gmail.com",
3   "coin_name": "colocoins",
4   "difference_percentage": 1
5 }
```

Cookies Headers (10) Test Results Status: 200 OK Time: 40 ms Size: 410 B Save as Example

Pretty Raw Preview Visualize JSON

```

1 {
2   "message": {
3     "email": "amirmohamad.babaei79@gmail.com",
4     "coin_name": "colocoins",
5     "percentage": 1
6   }
7 }
```

```

POST | http://127.0.0.1:8001/price/
Params Authorization Headers (8) Body Pre-request Script Tests Settings
none form-data x-www-form-urlencoded raw binary GraphQL JSON
1 ... "coin_name": "bitcoin"
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

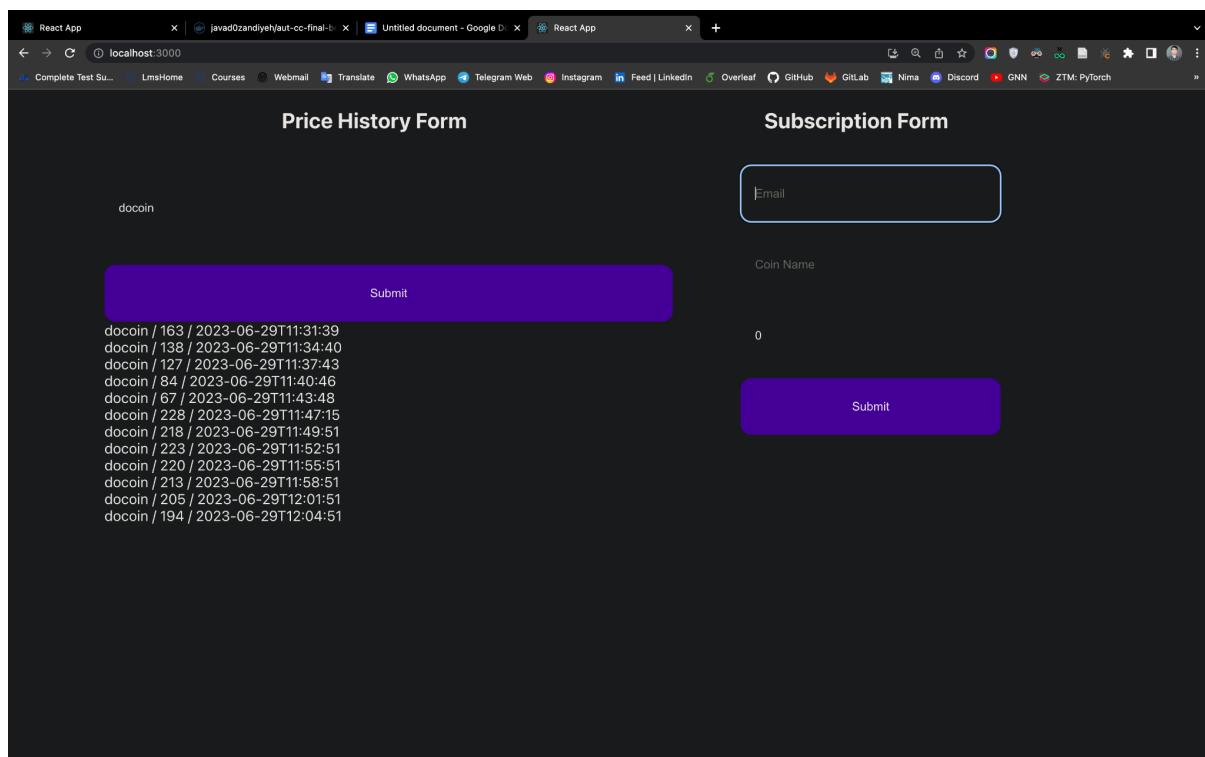
```

Status: 200 OK Time: 33 ms Size: 15.27 KB Save as Example

```

"message": [
    {
        "coin_name": "bitcoin",
        "time_stamp": "2023-06-29T15:09:15",
        "price": 49992.0
    },
    {
        "coin_name": "bitcoin",
        "time_stamp": "2023-06-29T15:09:18",
        "price": 49992.0
    },
    {
        "coin_name": "bitcoin",
        "time_stamp": "2023-06-29T15:09:21",
        "price": 49992.0
    },
    {
        "coin_name": "bitcoin",
        "time_stamp": "2023-06-29T15:09:24",
        "price": 49992.0
    },
    {
        "coin_name": "bitcoin",
        "time_stamp": "2023-06-29T15:09:27",
        "price": 49992.0
    }
]

```



۴. محتويات داکرفایل

در ساخت ایمیج ها از تکنیک multistage build استفاده شده است که حجم و سرعت کار را بالا ببریم:

```
(service_bepa) Dockerfile > Dockerfile > ...
You, 2 days ago | 2 authors (You and others)
1 FROM python:3.9.17-alpine AS builder
2
3 ENV PYTHONDONTWRITEBYTECODE=1
4 ENV PYTHONUNBUFFERED=1
5
6 WORKDIR /app
7 RUN apk --update add build-base mysql-dev bash
8 COPY requirements.txt .
9 RUN pip install -r requirements.txt
10
11 FROM python:3.9.17-alpine AS final
12
13 ENV PYTHONDONTWRITEBYTECODE=1
14 ENV PYTHONUNBUFFERED=1
| You, 2 days ago • Multistage build for docker files ...
15 COPY --from=builder /usr/local/lib/python3.9/site-packages /usr/local/lib/python3.9/site-packages
16
17 WORKDIR /app
18
19
20 CMD ["python", "bepa.py"]
```

```
(service_peyk) Dockerfile service_bepa > Dockerfile service_peyk > ...
You, 2 days ago | 1 author (You)
1 FROM python:3.9.17-alpine AS builder
2
3 ENV PYTHONDONTWRITEBYTECODE=1
4 ENV PYTHONUNBUFFERED=1
5
6 WORKDIR /app
7
8 COPY requirements.txt .
9
10 RUN apk --update add build-base mysql-dev bash
11 RUN pip3 install -r requirements.txt
12
13 FROM python:3.9.17-alpine AS final You, 2 days ago • Multistage build for docker files ...
14
15 ENV PYTHONDONTWRITEBYTECODE=1
16 ENV PYTHONUNBUFFERED=1
17
18 COPY --from=builder /usr/local/lib/python3.9/site-packages /usr/local/lib/python3.9/site-packages
19 COPY --from=builder /usr/lib/libmariadb.so.3 /usr/lib/
20 COPY --from=builder /usr/lib/mariadb/plugin/caching_sha2_password.so /usr/lib/mariadb/plugin/
21
22 EXPOSE ${APP_PORT}
23
24 WORKDIR /app/project_peyk
25
26 ENTRYPOINT ["/bin/sh", "entrypoint.sh"]
```

```
(service_react) Dockerfile service_bepa > Dockerfile service_react > ...
You, 5 days ago | 1 author (You)
1 FROM node:lts-alpine3.18
2
3 WORKDIR /app/app_react
4
5 COPY /app_react/package*.json /app/app_react
6
7 RUN npm install
8
9 ENTRYPOINT ["/bin/sh", "entrypoint.sh"]
10 |
```

۵. با استفاده از دستور `kubectl get` صحت ایجاد منابع بر روی کلaster را نمایش دهید

^۶. آدرس آیپی، ۲۱ بیادها و نحوه برقراری ارتباط میان آنها و سرویس ساخته شده

```
● [ ] ~ /De/f/u/c/l/h/f/AUT-CC-final-project/deployments | ⌂ main !1
● kubectl get all
NAME                                     READY   STATUS    RESTARTS   AGE
pod/bepa-service-bepa-58c7f4485d-klnrs   1/1     Running   0          15m
pod/coin-news-service-coin-news-5874fcbd54-zpzwj   1/1     Running   0          15m
pod/mysql-db-6995b88447-h49wq   1/1     Running   0          16m
pod/peyk-service-peyk-754f9d4bd5-2ftql   1/1     Running   0          15m

NAME           TYPE        CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
service/bepa-service-bepa   ClusterIP   10.100.5.171   <none>       8002/TCP   15m
service/coin-news-service-coin-news   ClusterIP   10.111.169.186   <none>       8000/TCP   15m
service/kubernetes   ClusterIP   10.96.0.1      <none>       443/TCP    53d
service/mysql-db   NodePort    10.99.244.153   <none>       3306:30249/TCP 16m
service/peyk-service-peyk   LoadBalancer  10.106.150.126  127.0.0.1     8001:31556/TCP 15m

NAME           READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/bepa-service-bepa   1/1      1           1           15m
deployment.apps/coin-news-service-coin-news   1/1      1           1           15m
deployment.apps/mysql-db   1/1      1           1           16m
deployment.apps/peyk-service-peyk   1/1      1           1           15m

NAME           DESIRED  CURRENT  READY   AGE
replicaset.apps/bepa-service-bepa-58c7f4485d   1        1        1        15m
replicaset.apps/coin-news-service-coin-news-5874fcbd54   1        1        1        15m
replicaset.apps/mysql-db-6995b88447   1        1        1        16m
replicaset.apps/peyk-service-peyk-754f9d4bd5   1        1        1        15m
```

در اجرای پروژه بر روی کلاستر باید به چند نکته توجه کرد:

الف) برای ارسال درخواست از خارج از کلاس `tunner` است که تونل زده شود با استفاده از `minikube`

ب) برای اینکه از فضای shared داخل لوکال استفاده شود نیاز است که برای هریک از پروژه های bepa , peyk , react به صورت

جدا گانه فضای داخل لوکال به فضای کلستر مپ شود با استفاده از minikube mount

پ) قسمت کوربنتیز با استفاده از بخش امتیازی helm زده شده است و دستورات `helm create`, `helm install`, `helm delete`

به کار برده خواهد شد

ج) برای لود کردن ایمیج های لوکال بر روی کوبرنتیز نیاز است که از دستور `minikube image load` استفاده شود

```
minikube tunnel
Tunnel successfully started

NOTE: Please do not close this terminal as this process must stay alive for the tunnel to be accessible ...

The service/ingress coin-news-service-coin-news requires privileged ports to be exposed: [80 443]
sudo permission will be asked for it.

Starting tunnel for service coin-news-service-coin-news.

Password:
Starting tunnel for service peyk-service-peyk.
Starting tunnel for service react-service-react.
Stopping tunnel for service react-service-react.
Starting tunnel for service react-service-react.
Stopping tunnel for service react-service-react.
Starting tunnel for service react-service-react.
Stopping tunnel for service react-service-react.
Starting tunnel for service react-service-react.
Stopping tunnel for service react-service-react.
Starting tunnel for service react-service-react.
Stopping tunnel for service react-service-react.
Starting tunnel for service react-service-react.
Stopping tunnel for service react-service-peyk.
Stopping tunnel for service coin-news-service-coin-news.
```

```
[-] ⚡ ~/De/f/u/cloud-computing/home-works/final-project/AUT-CC-final-project | ⚡ p main !1
  ✓ | base ♫ | 18:24:36 Ⓛ
  ↳ minikube mount $(pwd)/service_bepa:/app
    └─ Mounting host path /Users/javad/Desktop/files/university-courses/cloud-computing/home-works/final-project/AUT-CC-final-project/service_bepa into VM as /app/ ...
      └─ Mount type: 9p
      └─ User ID: docker
      └─ Group ID: docker
      └─ Version: 9p2000.L
      └─ Message Size: 262144
      └─ Options: map[]
      └─ Bind Address: 127.0.0.1:49675
    🌈 Userspace file server: ufs starting
  ✓ Successfully mounted /Users/javad/Desktop/files/university-courses/cloud-computing/home-works/final-project/AUT-CC-final-project/service_bepa to /app/
  ✅ NOTE: This process must stay alive for the mount to be accessible ...
```

```
[-] ⚡ ~/De/f/u/cl/home-works/final-project/AUT-CC-final-project | ⚡ p main !3           ✓ | base ♫ | 18:42:38 Ⓛ
  ↳ minikube mount $(pwd)/service_peyk/project_peyk:/app/project_peyk
    └─ Mounting host path /Users/javad/Desktop/files/university-courses/cloud-computing/home-works/final-project/AUT-CC-final-project/service_peyk/project_peyk into VM as /app/project_peyk ...
      └─ Mount type: 9p
      └─ User ID: docker
      └─ Group ID: docker
      └─ Version: 9p2000.L
      └─ Message Size: 262144
      └─ Options: map[]
      └─ Bind Address: 127.0.0.1:50039
    🌈 Userspace file server: ufs starting
  ✓ Successfully mounted /Users/javad/Desktop/files/university-courses/cloud-computing/home-works/final-project/AUT-CC-final-project/service_peyk/project_peyk to /app/project_peyk
  ✅ NOTE: This process must stay alive for the mount to be accessible ...
```

```
[-] ⚡ ~/De/f/u/cl/home-works/final-project/AUT-CC-final-project | ⚡ p main           ✓ | base ♫ | 19:47:44 Ⓛ
  ↳ minikube mount $(pwd)/service_react/app_react:/app/app_react
    └─ Mounting host path /Users/javad/Desktop/files/university-courses/cloud-computing/home-works/final-project/AUT-CC-final-project/service_react/app_react into VM as /app/app_react ...
      └─ Mount type: 9p
      └─ User ID: docker
      └─ Group ID: docker
      └─ Version: 9p2000.L
      └─ Message Size: 262144
      └─ Options: map[]
      └─ Bind Address: 127.0.0.1:51081
    🌈 Userspace file server: ufs starting
  ✓ Successfully mounted /Users/javad/Desktop/files/university-courses/cloud-computing/home-works/final-project/AUT-CC-final-project/service_react/app_react to /app/app_react
  ✅ NOTE: This process must stay alive for the mount to be accessible ...
```

دستور `minikube dashboard` هم برای نمایش پادها و سرویس‌ها به صورت web استفاده می‌شود

```

● └─ Apple | ➜ ~/De/f/u/cl/home-works/final-project/AUT-CC-final-project | ⚡ p main          ✓ | ✨ | base ✨ | 19:48:46 ✨
  minikube dashboard &
[2] 45946

○ └─ Apple | ➜ ~/De/f/u/cl/home-works/final-project/AUT-CC-final-project | ⚡ p main          ✓ | ✨ | base ✨ | 19:48:51 ✨
  🎉 Verifying dashboard health ...
  🎉 Launching proxy ...
  🎉 Verifying proxy health ...
  🎉 Opening http://127.0.0.1:51094/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/
  in your default browser...

```

دستورات زیر را به ترتیب اجرا میکنیم (در دایرکتوری که کانفیگ های هلم وجود دارد):

```

1 helm install mysql-db mysql_db/ --values mysql_db/values.yaml
2 helm install coin-news service_coin_news/ --values service_coin_news/values.yaml
3 helm install bepa service_bepa/ --values service_bepa/values.yaml
4 helm install peyk service_peyk/ --values service_peyk/values.yaml
5 helm install react service_react/ --values service_react/values.yaml
6

```

۷. برای دیپلومنت مربوط به پایگاه داده چه تعداد پاد ایجاد کردید؟ دلیل کار خود را توضیح دهید.

ریپلیکا را به صورت تستی یک گذاشته ایم اما با توجه به بخش امتیازی که پیاده کرده ایم (HPA) عملیات auto scaling را میتواند صورت گیرد و تعداد پادها به نسبت دلخواه افزایش یابد.

۸. به کمک یک پاد با ایمیج کل 22 و یا قابلیت port-forwarding کوئربندیز، سیستم نهایی خود را بیاماید. تصاویر صحت عملکرد درخواستهای `get_price_history` و `subscribe_coin` را بیاورید

عملیات tunneling را انجام میدهیم و نتایج به صورت زیر است:

The screenshot shows a Postman request configuration and its response. The request is a POST to `http://127.0.0.1:8001/subscribe/`. The Body tab is selected, showing a JSON payload:

```

1 {
2   "email": "amirmohamad.babaei79@gmail.com",
3   "coin_name": "colocoins",
4   "difference_percentage": 1
5 }

```

The response tab shows the following JSON data:

```

1 {
2   "message": {
3     "email": "amirmohamad.babaei79@gmail.com",
4     "coin_name": "colocoins",
5     "percentage": 1
6   }
7 }

```

Details from the response header: Status: 200 OK, Time: 40 ms, Size: 410 B, Save as Example.

```

POST | http://127.0.0.1:8001/price/
Params Authorization Headers (8) Body Pre-request Script Tests Settings
none form-data x-www-form-urlencoded raw binary GraphQL JSON
1 ...
2 ... "coin_name": "bitcoin"
3 ...

```

Body Cookies Headers (10) Test Results Status: 200 OK Time: 33 ms Size: 15.27 KB Save as Example

Pretty Raw Preview Visualize JSON

```

1 "message": [
2   {
3     "coin_name": "bitcoin",
4     "time_stamp": "2023-06-29T15:09:15",
5     "price": 49992.0
6   },
7   {
8     "coin_name": "bitcoin",
9     "time_stamp": "2023-06-29T15:09:18",
10    "price": 49992.0
11  },
12  {
13    "coin_name": "bitcoin",
14    "time_stamp": "2023-06-29T15:09:21",
15    "price": 49992.0
16  },
17  {
18    "coin_name": "bitcoin",
19    "time_stamp": "2023-06-29T15:09:24",
20    "price": 49992.0
21  },
22  {
23    "coin_name": "bitcoin",
24    "time_stamp": "2023-06-29T15:09:27",
25    "price": 49992.0
26  }
27 ]

```

موارد امتیازی پیاده سازی شده:

۱. ساخت HPA به منظور مقیاسسازی خودکار پادهای سرویس پیک.

در فایل values میتوان با استفاده از true کردن فلگ مورد نظر عملیات مدیریت pod ها را به صورت auto scaling هندل کرد.

```

autoscaling:
  enabled: false
  minReplicas: 1
  maxReplicas: 100
  targetCPUUtilizationPercentage: 80
  # targetMemoryUtilizationPercentage: 80

```

در شکل زیر معلوم است که از پارامترهای cpu و memory برای خودکار سازی پادها استفاده شده است و بعلاوه min , max نیز برای تعداد رپلیکا ها مشخصی شده است. همه موارد قابل کانفیگ کردن هستند. مهم ترین پارامتر های cpu , memory هستند زیرا باید کلاستر ظرفیت کافی برای ایجاد رپلیکا و همچنین قدرت پردازشی لازم برای مدیریت کردن آنها را داشته باشد. مقادیر این کانفیگ ها در فایل values قرار داده میشود.

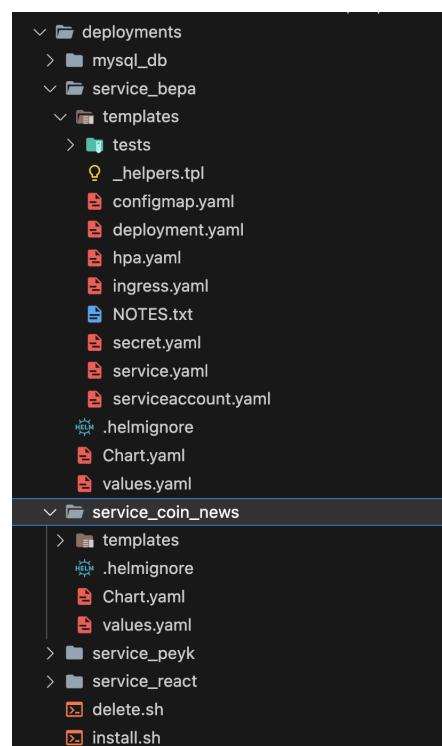
```

hpa.yaml ×
deployments > service_bepa > templates > hpa.yaml > metadata > name > {} {{ include "service_bepa.fullname" .}}
1   {{- if .Values.autoscaling.enabled -}}
2     apiVersion: autoscaling/v2
3     kind: HorizontalPodAutoscaler
4     metadata:
5       name: {{ include "service_bepa.fullname" . }} AmirMohammad Babaei, 2 days ago • feat: add helm ch
6       labels:
7         {{- include "service_bepa.labels" . | nindent 4 -}}
8     spec:
9       scaleTargetRef:
10         apiVersion: apps/v1
11         kind: Deployment
12         name: {{ include "service_bepa.fullname" . }}
13       minReplicas: {{ .Values.autoscaling.minReplicas }}
14       maxReplicas: {{ .Values.autoscaling.maxReplicas }}
15     metrics:
16       {{- if .Values.autoscaling.targetCPUUtilizationPercentage -}}
17       - type: Resource
18         resource:
19           name: cpu
20           target:
21             type: Utilization
22             averageUtilization: {{ .Values.autoscaling.targetCPUUtilizationPercentage }}
23       {{- end -}}
24       {{- if .Values.autoscaling.targetMemoryUtilizationPercentage -}}
25       - type: Resource
26         resource:
27           name: memory
28           target:
29             type: Utilization
30             averageUtilization: {{ .Values.autoscaling.targetMemoryUtilizationPercentage }}
31       {{- end -}}
32 {{- end -}}

```

۲. پیاده سازی چارت هم

در طول گزارش به آن اشاره شد، شما می‌ایز فایل‌ها به صورت زیر است



ساختار هلم چارت به این صورت است که به صورت خام دارای فایل های `chart.yaml`, `values.yaml` و `templates` و .. است. در فایل `values` مقادیر قابل کانفیگ قرار داده می شود (مثلاً اینکه `shared volume` در کجا باشد و یا موارد دیگری در قسمت قبل توضیح داده شد). در دایرکتوری `templates` فایل هایی برای ایجاد `deployment`, `secret`, `hpa`, `service`, ... قرار دارند که به صورت تمپلیت هستند و مقادیر خود را به کمک `Values` از فایل `values` می خوانند.

پارامترهای `values` میتوانند هرچیزی که بخواهیم استفاده کنیم در تمپلیت ها باشند اما موارد زیر به عنوان مثال در این پروژه استفاده شده اند:

```
# Default values for service_peyk.
# This is a YAML-formatted file.
# Declare variables to be passed into your templates.

replicaCount: 1

image:
repository: peyk
pullPolicy: IfNotPresent
# Overrides the image tag whose default is the chart appVersion.
tag: "v1"

imagePullSecrets: []
nameOverride: ""
fullnameOverride: ""

serviceAccount:
# Specifies whether a service account should be created
create: true
# Annotations to add to the service account
annotations: {}
# The name of the service account to use.
# If not set and create is true, a name is generated using the fullname template
name: "service-peyk"

podAnnotations: {}

podSecurityContext: {}
# fsGroup: 2000

securityContext: {}
# capabilities:
# drop:
# - ALL
# readOnlyRootFilesystem: true
# runAsNonRoot: true
# runAsUser: 1000

service:
type: LoadBalancer
port: 8001
targetPort: 8001
```

```
ingress:
  enabled: false
  className: ""
  annotations: {}
  # kubernetes.io/ingress.class: nginx
  # kubernetes.io/tls-acme: "true"
  hosts:
    - host: chart-example.local
  paths:
    - path: /
  pathType: ImplementationSpecific
  tls: []
  # - secretName: chart-example-tls
  # hosts:
  # - chart-example.local

resources: {}
# We usually recommend not to specify default resources and to leave this as a conscious
# choice for the user. This also increases chances charts run on environments with little
# resources, such as Minikube. If you do want to specify resources, uncomment the following
# lines, adjust them as necessary, and remove the curly braces after 'resources:'.
# limits:
#   cpu: 100m
#   memory: 128Mi
# requests:
#   cpu: 100m
#   memory: 128Mi

autoscaling:
  enabled: false
  minReplicas: 1
  maxReplicas: 100
  targetCPUUtilizationPercentage: 80
  # targetMemoryUtilizationPercentage: 80

nodeSelector: {}

tolerations: []

affinity: {}

configs:
  APP_IP: 0.0.0.0
  APP_PORT: 8001
  MYSQL_DATABASE: coin_db
  MYSQL_HOST: mysql-db
  MYSQL_PORT: 3306
  MYSQL_USER: admin

secrets:
  MYSQL_ROOT_PASSWORD: pass

volumes:
  name: peyk-volume
  hostPath: /app/project_peyk
  mountPath: /app/project_peyk
```

۳. پیاده سازی داکر کامپوز جهت خودکارسازی ایجاد منابع و وابستگی های مورد نیاز پروژه و نهایتاً build و اجرای آن.

در طول گزارش با آن اشاره شد و فایل هم به صورت زیر است:

```
docker-compose.yml ×
dockercospe.yml > {} services > {} react > [ ] ports
  docker-compose.yml - The Compose specification establishes a standard for the definition of multi-container platform-agnostic applications. ...
1 version: "3.8"
2
3 services:
4   mysql_db:
5     image: mysql:8.0
6     container_name: mysql_db
7     restart: always
8     env_file:
9       - ./env/shared.env'
10    ports:
11      - 3306:3306
12    volumes:
13      - ./service_mysql_db/data:/var/lib/mysql
14      - ./service_mysql_db/script:/docker-entrypoint-initdb.d
15
16  coin_news:
17    image: amirhossein21/coinnews.macos:v0.2
18    container_name: coin_news
19    restart: always
20    ports:
21      - 8000:8000
22    volumes:
23      - type: bind
24        source: ./service_coin_news/config.yml
25        target: /go/src/app/config.yml
26
27
28  bepa:
29    image: bepa:v1
30    build:
31      context: ./service_bepa
32      dockerfile: Dockerfile
33    container_name: bepa
34    restart: always
35    ports:
36      - 8002:8002
37    volumes:
38      - ./service_bepa:/app
39    environment:
40      - INTERVAL_SECONDS=180
41      - COINNEWS_ENDPOINT=http://coin_news:8000
42    env_file:
43      - ./env/shared.env'
44    depends_on:
45      - mysql_db
46      - coin_news
```

```

47  peyk:
48    image: peyk:v1
49    build:
50      context: ./service_peyk
51      dockerfile: Dockerfile
52    container_name: peyk
53    restart: always
54    ports:
55      - 8001:8001
56    volumes:
57      - ./service_peyk/project_peyk:/app/project_peyk
58    working_dir: /app/project_peyk
59    environment:
60      - APP_IP=0.0.0.0
61      - APP_PORT=8001
62    env_file:
63      - './env/shared.env'
64    depends_on:
65      - mysql_db
66
67 react:
68   image: react:v1
69   build:
70     context: ./service_react
71     dockerfile: Dockerfile
72   container_name: react
73   restart: always
74   ports: You, 5 days ago • Add react service ...
75     - 3000:3000
76   volumes:
77     - ./service_react/app_react:/app/app_react

```

در داکر کامپوز نیز مدیرت ساخت ایمیج و کانتینر به صورت خودکار انجام میشود. به عنوان مثال اینکه `shared voluem` یک کانتینر

چیست. این کانتینر از روی کدام داکر فایل قرار است ساخته شود. اسم کانتینر چه چیزی است. در کدام `network` قرار دارد. به کدام

کانتینر ها پیش از ساخته شدن نیاز دارد (وابسطه است). مقادیر `env` مورد استفاده در کانتینر چیست و ...