

بسمه تعالی

تمرین دوم مبانی رایانش ابری

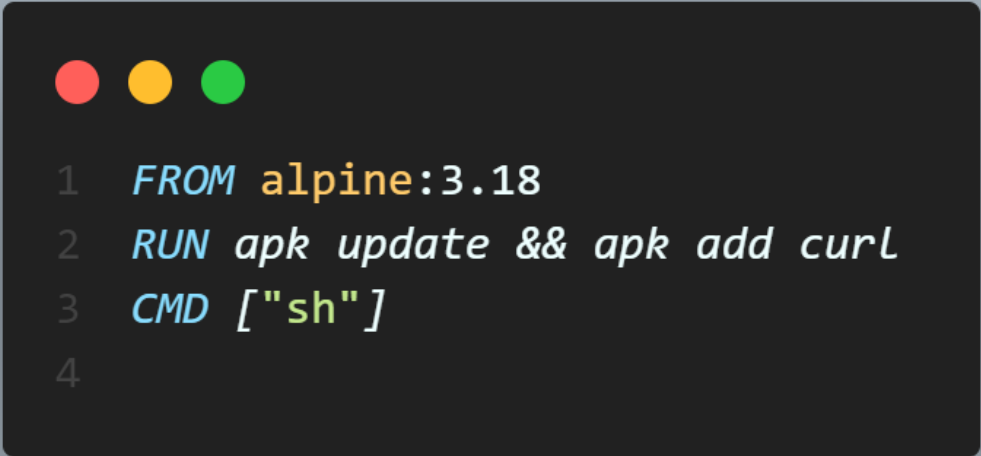
علیرضا زارع زین آبادی 9931022

ترم پاییز 1402

قسمت اول تمرین (داکر):

بخش اول:

Alpine + cURL



```
1 FROM alpine:3.18
2 RUN apk update && apk add curl
3 CMD ["sh"]
4
```

با بیلد کردن این داکر فایل به یک ایمیج الپاین دارای cURL میرسیم.

اسکرین شات های مراحل ساخت؛ تست؛ و قراردعی روی داکرهاب از صفحه بعد.

```
ahora@srv9760378373: ~/cloud/dockerhw/FirstStep
ahora@srv9760378373:~/cloud/dockerhw/FirstStep$ nano Dockerfile
ahora@srv9760378373:~/cloud/dockerhw/FirstStep$ docker build -t ahora-alpine .
[+] Building 3.0s (7/7) FINISHED                                docker:default
=> [internal] load build definition from Dockerfile              0.0s
=> => transferring dockerfile: 96B                               0.0s
=> [internal] load .dockerignore                                0.0s
=> => transferring context: 2B                                    0.0s
=> [internal] load metadata for docker.io/library/alpine:3.18   1.4s
=> [auth] library/alpine:pull token for registry-1.docker.io   0.0s
=> [1/2] FROM docker.io/library/alpine:3.18@sha256:eece025e432126ce23f22 0.3s
=> => resolve docker.io/library/alpine:3.18@sha256:eece025e432126ce23f22 0.0s
=> => sha256:48d9183eb12a05c99bcc0bf44a003607b8e941e1d4f41f9 528B / 528B 0.0s
=> => sha256:8ca4688f4f356596b5ae539337c9941abc78eda1002 1.47kB / 1.47kB 0.0s
=> => sha256:96526aa774ef0126ad0fe9e9a95764c5fc37f409ab9 3.40MB / 3.40MB 0.2s
=> => sha256:eece025e432126ce23f223450a0326fbebde39cdf49 1.64kB / 1.64kB 0.0s
=> => extracting sha256:96526aa774ef0126ad0fe9e9a95764c5fc37f409ab9e9702 0.1s
=> [2/2] RUN apk update && apk add curl                        1.1s
=> exporting to image                                           0.1s
=> => exporting layers                                           0.1s
=> => writing image sha256:5a918776485405514a10ea84d339b4f72cbb1a498f3e9 0.0s
=> => naming to docker.io/library/ahora-alpine                  0.0s
ahora@srv9760378373:~/cloud/dockerhw/FirstStep$
```

```
ahora@srv9760378373: ~/cloud/dockerhw/FirstStep
=> => transferring context: 2B                                    0.0s
=> [internal] load metadata for docker.io/library/alpine:3.18   1.4s
=> [auth] library/alpine:pull token for registry-1.docker.io   0.0s
=> [1/2] FROM docker.io/library/alpine:3.18@sha256:eece025e432126ce23f22 0.3s
=> => resolve docker.io/library/alpine:3.18@sha256:eece025e432126ce23f22 0.0s
=> => sha256:48d9183eb12a05c99bcc0bf44a003607b8e941e1d4f41f9 528B / 528B 0.0s
=> => sha256:8ca4688f4f356596b5ae539337c9941abc78eda1002 1.47kB / 1.47kB 0.0s
=> => sha256:96526aa774ef0126ad0fe9e9a95764c5fc37f409ab9 3.40MB / 3.40MB 0.2s
=> => sha256:eece025e432126ce23f223450a0326fbebde39cdf49 1.64kB / 1.64kB 0.0s
=> => extracting sha256:96526aa774ef0126ad0fe9e9a95764c5fc37f409ab9e9702 0.1s
=> [2/2] RUN apk update && apk add curl                        1.1s
=> exporting to image                                           0.1s
=> => exporting layers                                           0.1s
=> => writing image sha256:5a918776485405514a10ea84d339b4f72cbb1a498f3e9 0.0s
=> => naming to docker.io/library/ahora-alpine                  0.0s
ahora@srv9760378373:~/cloud/dockerhw/FirstStep$ docker run -it ahora-alpine
/ # 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>
/ #
```

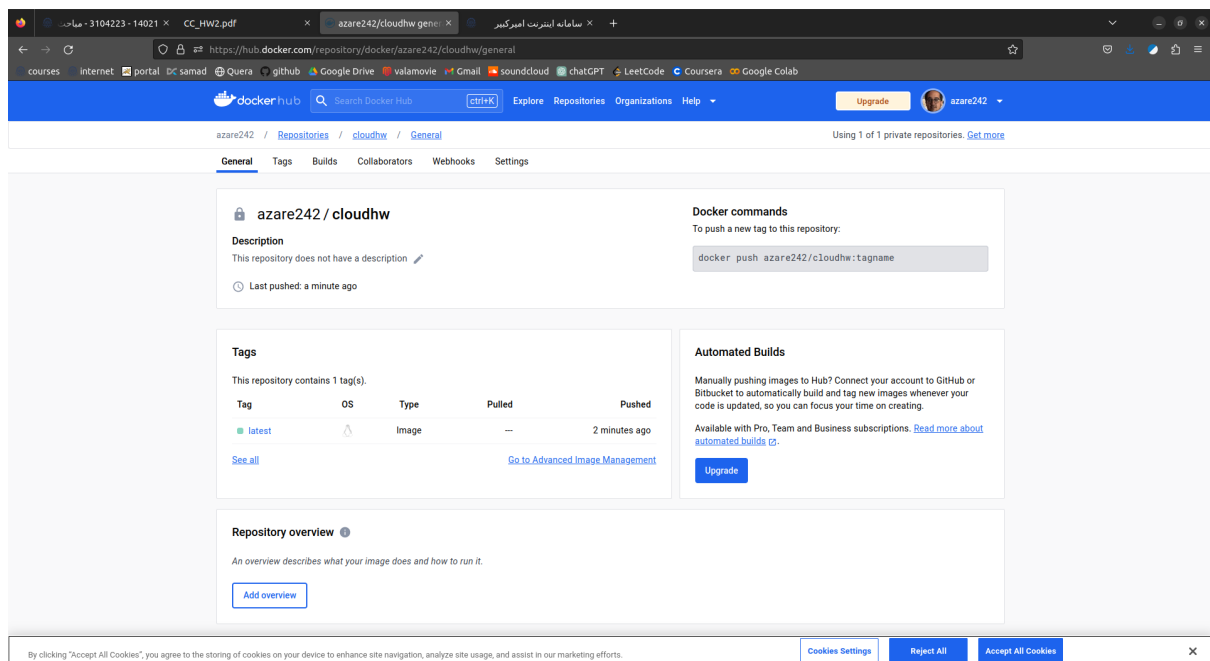
```
ahora@srv9760378373: ~/cloud/dockerhw/FirstStep
ahora@srv9760378373:~/cloud/dockerhw/FirstStep$ docker push ahora-alpine
Using default tag: latest
The push refers to repository [docker.io/library/ahora-alpine]
5db99d9b5a75: Preparing
cc2447e1835a: Preparing
denied: requested access to the resource is denied
ahora@srv9760378373:~/cloud/dockerhw/FirstStep$ docker push ahora-alpine azare242/cloudhw
"docker push" requires exactly 1 argument.
See 'docker push --help'.

Usage:  docker push [OPTIONS] NAME[:TAG]

Upload an image to a registry
ahora@srv9760378373:~/cloud/dockerhw/FirstStep$ docker tag ahora-alpine azare242/cloudhw
ahora@srv9760378373:~/cloud/dockerhw/FirstStep$ docker push azare242/cloudhw
Using default tag: latest
The push refers to repository [docker.io/azare242/cloudhw]
5db99d9b5a75: Pushed
cc2447e1835a: Pushed
latest: digest: sha256:5a18e7cf4de7c9bc563a2cbb80f3cd4d42a4055c8b40f4288d9c7097c9c7f8bb size: 738
ahora@srv9760378373:~/cloud/dockerhw/FirstStep$
```

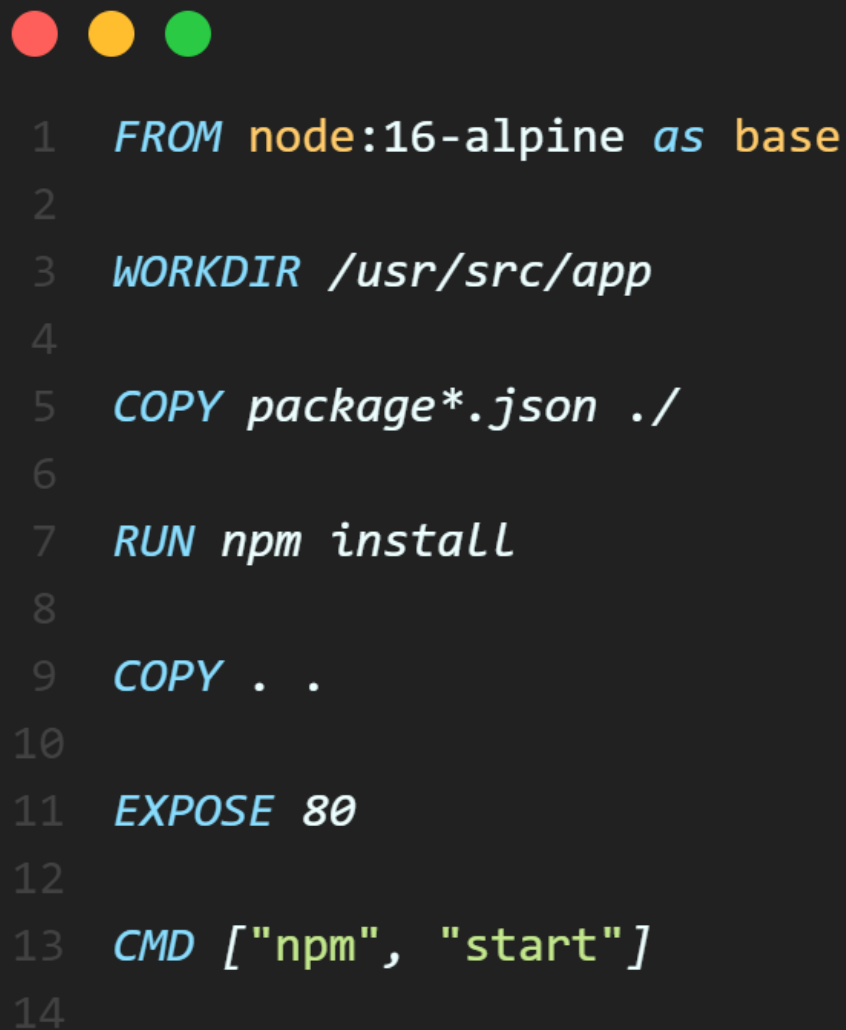
```
ahora@srv9760378373: ~/cloud/dockerhw/FirstStep
ahora@srv9760378373:~/cloud/dockerhw/FirstStep$ docker image ls
REPOSITORY          TAG         IMAGE ID      CREATED       SIZE
azare242/cloudhw    latest      5a9187764854  28 minutes ago  13.4MB
ahora-alpine        latest      5a9187764854  28 minutes ago  13.4MB
ahora@srv9760378373:~/cloud/dockerhw/FirstStep$
```

```
ahora@srv9760378373: ~/cloud/dockerhw/FirstStep
azare242/cloudhw latest 5a9187764854 28 minutes ago 13.4MB
ahora-alpine latest 5a9187764854 28 minutes ago 13.4MB
ahora@srv9760378373:~/cloud/dockerhw/FirstStep$ docker push azare242/cloudhw
Using default tag: latest
The push refers to repository [docker.io/azare242/cloudhw]
5db99d9b5a75: Layer already exists
cc2447e1835a: Layer already exists
latest: digest: sha256:5a18e7cf4de7c9bc563a2cbb80f3cd4d42a4055c8b40f4288d9c7097c9c7f8bb size: 738
ahora@srv9760378373:~/cloud/dockerhw/FirstStep$ docker image ls
REPOSITORY TAG IMAGE ID CREATED SIZE
ahora-alpine latest 5a9187764854 28 minutes ago 13.4MB
azare242/cloudhw latest 5a9187764854 28 minutes ago 13.4MB
ahora@srv9760378373:~/cloud/dockerhw/FirstStep$ docker run -it azare242/cloudhw
/ # curl google.col
curl: (6) Could not resolve host: google.col
/ # 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>
/ #
```



بخش دوم: سرور هواشناسی.
این سرور با استفاده از پکیج Express و محیط Nodejs با زبان جاوااسکریپت توسعه داده شد.
/api/v1/weather/<CITY NAME>
اندپوینتی است که اطلاعات آب و هوا را برمیگرداند.

داکر فایل برای ساختن ایمج آن:

A dark-themed code editor window with three colored window control buttons (red, yellow, green) in the top-left corner. The editor contains a Dockerfile with 14 lines of code, each preceded by a line number from 1 to 14. The code defines a Docker image based on node:16-alpine, sets the working directory to /usr/src/app, copies package*.json files, installs npm dependencies, copies the application source code, exposes port 80, and sets the command to run npm start.

```
1 FROM node:16-alpine as base
2
3 WORKDIR /usr/src/app
4
5 COPY package*.json ./
6
7 RUN npm install
8
9 COPY . .
10
11 EXPOSE 80
12
13 CMD ["npm", "start"]
14
```

مراحل بیلد ایمیج:

```
Administrator: Windows PowerShell
PS E:\cc\ahora_weather\weather_server> docker build -t weather-server .
[+] Building 0.0s (0/0)
2023/11/28 21:59:14 http2: server: error reading preface from client ///pipe/docker_engine: file has already been clo
[+] Building 0.0s (0/0)
```

```

Administrator: Windows PowerShell
=> [1/5] FROM docker.io/library/node:16-alpine@sha256:a1f9d027912b58a7c75be7716c97cfbc6d3099f3a97ed84aa490be9 76.7s
=> => resolve docker.io/library/node:16-alpine@sha256:a1f9d027912b58a7c75be7716c97cfbc6d3099f3a97ed84aa490be9d 2.2s
=> => sha256:a1f9d027912b58a7c75be7716c97cfbc6d3099f3a97ed84aa490be9dee20e787 1.43kB / 1.43kB 0.0s
=> => sha256:72e89a86be58922ed7b1475e5e6f151537676470695dd106521738b060e139d 1.16kB / 1.16kB 0.0s
=> => sha256:2573171e0124bb95d14d128728a52a977bb917ef45d7c4fa8cfe76bc44aa78b73 6.73kB / 6.73kB 0.0s
=> => sha256:7264a8db6415046d36d16ba98b79778e18accee6ffa71850405994cffa9be7de 3.40MB / 3.40MB 7.3s
=> => sha256:eee371b9ce3ffdbb8aa703b9a14d318801ddc3468f096bb6cfeabbeb715147f9 36.63MB / 36.63MB 22.5s
=> => sha256:93b3025fe10392717d06ec0d012a9ffa2039d766a322aac899c6831dd93382c2 2.34MB / 2.34MB 6.7s
=> => extracting sha256:7264a8db6415046d36d16ba98b79778e18accee6ffa71850405994cffa9be7de 1.8s
=> => sha256:d9059661ce70092af66d2773666584fc8addcb78a2be63f720022f4875577ea9 452B / 452B 9.7s
=> => extracting sha256:eee371b9ce3ffdbb8aa703b9a14d318801ddc3468f096bb6cfeabbeb715147f9 17.9s
=> => extracting sha256:93b3025fe10392717d06ec0d012a9ffa2039d766a322aac899c6831dd93382c2 1.8s
=> => extracting sha256:d9059661ce70092af66d2773666584fc8addcb78a2be63f720022f4875577ea9 0.0s
=> [internal] load build context 6.0s
=> => transferring context: 4.75kB 2.1s
=> [2/5] WORKDIR /usr/src/app 22.5s
=> [3/5] COPY package*.json ./ 3.0s
=> [4/5] RUN npm install 47.7s
=> [5/5] COPY . . 2.7s
=> exporting to image 9.3s
=> => exporting layers 8.7s
=> => writing image sha256:b4b677ba423f185b45b32e27ebfeed1d5462b4ac76f2c1fa45d0ccc8891d2168 0.2s
=> => naming to docker.io/library/weather-server 0.4s

What's Next?
View a summary of image vulnerabilities and recommendations -> docker scout quickview
PS E:\cc\ahora weather\weather server>

```

Administrator: Windows PowerShell				
REPOSITORY	IMAGE ID	CREATED	SIZE	TAG
weather-server	b4b677ba423f	3 minutes ago	131MB	latest
hubproxy.docker.internal:5555/docker/desktop-kubernetes	0.3.4-1-debian 1d7e8203bdb9	7 weeks ago	430MB	kubernetes-v1.28.2-cni-v1.3.0-critools-v1.28.0-cri-dockerd-v
registry.k8s.io/kube-apiserver	cdcab12b2dd1	2 months ago	126MB	v1.28.2
registry.k8s.io/kube-scheduler	7a5d9d67a13f	2 months ago	60.1MB	v1.28.2
registry.k8s.io/kube-controller-manager	55f13c92defb	2 months ago	122MB	v1.28.2
registry.k8s.io/kube-proxy	c120fed2beb8	2 months ago	73.1MB	v1.28.2
registry.k8s.io/etcd	73deb9a3f702	6 months ago	294MB	3.5.9-0
docker/desktop-vpnkit-controller	556098075b3d	6 months ago	36.2MB	dc331cb22850be0cdd97c84a9cfecaf44a1afb6e
registry.k8s.io/coredns/coredns	ead0a4a53df8	9 months ago	53.6MB	v1.10.1
registry.k8s.io/etcd	86b6af7dd652	10 months ago	296MB	3.5.7-0
registry.k8s.io/pause	e6f181688397	13 months ago	744kB	3.9
docker/desktop-storage-provisioner	99f89471f470	2 years ago	41.9MB	v2.0
PS E:\cc\ahora_weather\weather_server>				

قراردهی ایمیج روی داکر هاب:

```
Command Prompt
docker/desktop-storage-provisioner v2.0 99f89471f470 2 years ago 41.9MB

C:\Users\Lenovo>docker push azare242/weather-server:tagname
The push refers to repository [docker.io/azare242/weather-server]
An image does not exist locally with the tag: azare242/weather-server

C:\Users\Lenovo>docker tag --help
Usage: docker tag SOURCE_IMAGE[:TAG] TARGET_IMAGE[:TAG]
Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE

Aliases:
  docker image tag, docker tag

C:\Users\Lenovo>docker tag weather-server azare242/weather-server:latest

C:\Users\Lenovo>docker push azare242/weather-server:tagname
The push refers to repository [docker.io/azare242/weather-server]
tag does not exist: azare242/weather-server:tagname


C:\Users\Lenovo>docker image ls
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
azare242/weather-server latest             b4b677ba423f       30 minutes ago     131MB
weather-server      latest             b4b677ba423f       30 minutes ago     131MB
subproxy.docker.internal:5555/docker-desktop-kubernetes  v1.28.2            1d7e8203bdb9       7 weeks ago        430MB
registry.k8s.io/kube-apiserver      v1.28.2            cdcab12b2dd1       2 months ago       126MB
registry.k8s.io/kube-controller-manager v1.28.2            55f13c92de9b       2 months ago       122MB
registry.k8s.io/kube-scheduler      v1.28.2            7a549d67a13f       2 months ago       68.1MB
registry.k8s.io/kube-proxy           v1.28.2            c120fed2beb8       2 months ago       73.1MB
registry.k8s.io/etcd                 3.5.9-0            73deb9a3f702       6 months ago       294MB
docker/desktop-vpnkit-controller     v1.10.1            dc331cb22850be0cdd97c84a9cfecaf44a1afb6e 6 months ago       36.2MB
registry.k8s.io/coredns/coredns      v1.10.1            ead8a4a53df8       9 months ago       53.6MB
registry.k8s.io/etcd                 3.5.7-0            86b6af7dd652       10 months ago     296MB
registry.k8s.io/pause                 3.9                e6f181688397       13 months ago     744kB
docker/desktop-storage-provisioner   v2.0                99f89471f470       2 years ago        41.9MB


C:\Users\Lenovo>docker push azare242/weather-server:latest
The push refers to repository [docker.io/azare242/weather-server]
6668840287f6: Pushed
81b00f9d1153: Pushed
72385b3f9a08: Pushed
f11ee1232489: Pushed
365ccd918307: Pushed
1bba629c69e9: Pushed
139c1270acfl: Pushed
4693057ce236: Pushed
latest: digest: sha256:dc167f60e915e14e8837323885ce99512725107bde1b243427c06e4d7a350e20 size: 1991

C:\Users\Lenovo>
```

azare242 / weather-server

Description

This repository does not have a description 

 Last pushed: a few seconds ago

Tags

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
 latest		Image	---	a few seconds ago

[See all](#)

[Go to Advanced Image Management](#)

همچنین inspect ایمیج:

```
Command Prompt
C:\Users\Lenovo>kubectl get nodes
NAME                STATUS    ROLES    AGE   VERSION
docker-desktop      Ready    control-plane   4m22s   v1.28.2

C:\Users\Lenovo>kubectl get pods
No resources found in default namespace.

C:\Users\Lenovo>docker inspect weather-server
[
  {
    "Id": "sha256:b4b677ba423f185b45b32e27ebfeed1d5462b4ac76f2c1fa45d8ccc8891d22168",
    "RepoTags": [
      "weather-server:latest"
    ],
    "RepoDigests": [],
    "Parent": "",
    "Comment": "buildkit.dockerfile.v0",
    "Created": "2023-11-28T18:32:43.243026401Z",
    "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,
      "ExposedPorts": {
        "80/tcp": {}
      }
    }
  }
]
```

```
1 version: '3'
2 services:
3   app:
4     image: "azare242/weather-server:latest"
5     ports:
6       - "9999:80"
7     depends_on:
8       - redis
9     environment:
10      - PORT=80
11      - REDIS_HOST=redis
12      - REDIS_PORT=6379
13      - WEATHER_URL=https://api.api-ninjas.com/v1/weather?city=
14      - WEATHER_AUTH=YbdgZf59lSo6VSJ6BaklYQ==xghcOBjyCR8UbPPf
15      - REDIS_PAIR_TTL=300
16   redis:
17     image: "redis:7.2-alpine"
18     ports:
19       - "8989:6379"
20     volumes:
21       - redis-data:/data
22 volumes:
23   redis-data:
24
```

از ردیس 7.2 با بیس ایمیج الپاین استفاده شده است.

والیوم redis-data برای پرسپست کردن دیتای ردیس استفاده می شود.
کامپوز خودش نتورک مورد نیاز را میسازد.

```

weather_server-app-1 > weather@0.0.0 start
weather_server-app-1 > node index.js
weather_server-app-1 {
weather_server-app-1   PORT: '80',
weather_server-app-1   REDIS_HOST: 'redis',
weather_server-app-1   REDIS_PORT: '6379',
weather_server-app-1   REDIS_PAIR_TTL: '300',
weather_server-app-1   WEATHER_URL: 'https://api.api-ninjas.com/v1/weather?city=',
weather_server-app-1   WEATHER_AUTH: 'YbdgZf59lSo6VSJ6BaklYQ==xghcOBjyCR8UbPPf'
weather_server-app-1 }
weather_server-app-1 Server running on 80
weather_server-app-1 CACHE MISS
weather_server-app-1 [GET] "/api/v1/weather/tehran" (200) 31 - 3429.227 ms
weather_server-app-1 CACHE HIT
weather_server-app-1 [GET] "/api/v1/weather/tehran" (200) 31 - 5.237 ms
weather_server-app-1 CACHE HIT
weather_server-app-1 [GET] "/api/v1/weather/tehran" (200) 31 - 135.414 ms
weather_server-app-1 CACHE HIT
weather_server-app-1 [GET] "/api/v1/weather/tehran" (200) 31 - 12.714 ms
weather_server-app-1 CACHE MISS
weather_server-app-1 [GET] "/api/v1/weather/rafsanjan" (200) 31 - 2216.120 ms
weather_server-app-1 CACHE HIT
weather_server-app-1 [GET] "/api/v1/weather/rafsanjan" (200) 31 - 2.914 ms
weather_server-app-1 CACHE HIT
weather_server-app-1 [GET] "/api/v1/weather/rafsanjan" (200) 31 - 8.095 ms
weather_server-app-1 CACHE HIT
weather_server-app-1 [GET] "/api/v1/weather/rafsanjan" (200) 31 - 2.307 ms
weather_server-app-1 CACHE MISS
weather_server-app-1 [GET] "/api/v1/weather/ardabil" (200) 29 - 1163.215 ms

```

نتایج کش کردن را مشاهده میکنیم.
مقدار اعتبار هر Pair کش 5 دقیقه است.

قسمت دوم تمرین (کوبرنتیز)
از داکر برای نصب مینی کیوب استفاده شده است و نتایج نصب آن:

```
Command Prompt
C:\Users\Lenovo>kubectl get nodes
NAME          STATUS    ROLES          AGE    VERSION
docker-desktop Ready     control-plane  4m22s  v1.28.2

C:\Users\Lenovo>kubectl get pods
No resources found in default namespace.

C:\Users\Lenovo>
```

یسورس های مورد استفاده برای اجرای سرویس شامل Service Deployment ConfigMap هستند:



```
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: weather-server-service
5  spec:
6    selector:
7      app: weather-server
8    ports:
9      - protocol: TCP
10      port: 80
11      targetPort: 80
12
```



```
1  apiVersion: v1
2  kind: ConfigMap
3  metadata:
4    name: my-configmap
5  data:
6    config.yaml: -|
7      PORT: 80
8      REDIS_HOST: redis
9      REDIS_PORT: 6379
10     WEATHER_URL: https://api.api-ninjas.com/v1/weather?city=
11     WEATHER_AUTH: YbdgZf59lSo6VSJ6BaklYQ==xghcOBjyCR8UbPPf
12     REDIS_PAIR_TTL: 300
```



```
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: weather-server-deployment
5    labels:
6      app: weather-server
7  spec:
8    replicas: 3
9    selector:
10     matchLabels:
11       app: weather-server
12   template:
13     metadata:
14       labels:
15         app: weather-server
16     spec:
17       containers:
18         - name: wather-server
19           image: azare242/weather-server:latest
20           ports:
21             - containerPort: 80
22           volumeMounts:
23             - mountPath: /configs
24               name: config-volume
25       volumes:
26         - name: config-volume
27           configMap:
28             name: my-configmap
```

با دستور `kubectl apply -f` می‌توان آن‌ها را به گره خود اضافه کرد.

در اینجا چون تعداد رپلیکاهای سرور هواشناسی را 3 در نظر گرفتیم 3 پاد از ان ساخته می شود.

```
E:\cc\ahora_weather\deployment\k8s>kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
redis-7bdf47445b-s56rs	1/1	Running	0	82s
weather-server-deployment-79f5dff7c4-bpfdk	1/1	Running	0	77s
weather-server-deployment-79f5dff7c4-l7bq5	1/1	Running	0	77s
weather-server-deployment-79f5dff7c4-wlidsl	1/1	Running	0	77s

چون پاد در صورت از بین رفتن ایپی خود را از دست میدهد نیاز داریم که از ایپی سرویس استفاده کنیم. که سرویس و کلاستر ایپی درخواست هارا به پاد ها ارسال میکند. که فقط از داخل کلاستر قابل دستیابی است. برای دسترسی از خارج کلاستر باید از نود پورت یا لود بالانسر استفاده کرد. (در فایل های yml ما نوع سرویس را مشخص نکردیم چون مقدار پیشفرض از طرف کوبرنتیز کلاستر ایپی است)

حال نوبت به دیپلوی کردن ردیس است.



```
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: redis-cluster-service
5  spec:
6    type: ClusterIP
7    ports:
8      - targetPort: 6379
9        port: 6379
10       protocol: TCP
11    selector:
12      name: redis
```



```
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: redis
5  spec:
6    replicas: 1
7    selector:
8      matchLabels:
9        name: redis
10   template:
11     metadata:
12       labels:
13         name: redis
14     spec:
15       volumes:
16         - name: redis-data-store
17           persistentVolumeClaim:
18             claimName: redis-pv-claim
19       containers:
20         - name: redis
21           image: redis:7.2-alpine
22           ports:
23             - containerPort: 6379
24           volumeMounts:
25             - mountPath: "/data"
26               name: redis-data-store
27
```

با پورت فوروارد کردن با استفاده از دستور `kubectl port-forward` می توان به آن دسترسی داشت.

```
E:\cc\ahora_weather\deployment\k8s>kubectl port-forward service/weather-server-service 8000:80
Forwarding from 127.0.0.1:8000 -> 80
Forwarding from [::1]:8000 -> 80
```

حال اگر پاد ردیس از بین برود و دوباره آن را اجرا کنیم دیتای آن از دست می رود پس باید pv و pvc برای ردیس نوشت:

```
1 apiVersion: v1
2 kind: PersistentVolume
3 metadata:
4   name: redis-volume
5   labels:
6     type: local
7 spec:
8   storageClassName: manual
9   capacity:
10    storage: 1Gi
11   accessModes:
12    - ReadWriteOnce
13   hostPath:
14    path: "/home/dev/redis_data"
```

```
1 apiVersion: v1
2 kind: PersistentVolumeClaim
3 metadata:
4   name: redis-pv-claim
5 spec:
6   storageClassName: manual
7   accessModes:
8    - ReadWriteOnce
9   resources:
10    requests:
11     storage: 500Mi
12
```

مینی کیوب به ما dynamic provisioning را ارائه میکند. این قابلیت خودکار حافظه تخصیص میدهد. که باعث مدیریت بهتر در کلاستر ها می شود.

بصورت کلی هر pv را فقط میتوان به pvc های همان نود اختصاص داد و در صورتی که یک pvc نود بخواهد به آن دسترسی یابد باید pvc دیگری دسترسی نخواهد داشت مگر اینکه بقیه pvc ها خود را به pv بایند کنند.. این قابلیت به isolation فضای ذخیره سازی کمک میکند.

```
E:\cc\ahora_weather\deployment\k8s>curl localhost:8000/api/v1/weather/tehran
{"weather":{"min":8,"max":11}}
```

```
E:\cc\ahora_weather\deployment\k8s>kubectl logs weather-server-deployment-79f5dff7c4-wdxw6
> weather@0.0.0 start
> node index.js
{
  PORT: '80',
  REDIS_HOST: 'redis',
  REDIS_PORT: '6379',
  REDIS_PAIR_TTL: '300',
  WEATHER_URL: 'https://api.api-ninjas.com/v1/weather?city=',
  WEATHER_AUTH: 'YbdgZf59lSo6VSJ6BaklYQ==xghc0BjyCR8UbPPf'
}
Server running on 80
```

با **port-forward** کردن میتوان به سرویس دسترسی داشت که سرویس خود با انتخاب پاد به آن درخواست میزند.

حال چون **pV** برای ردیس داریم هر بار که پاد ردیس نابود شود و جایگزین شود دیتا همیشه پابرجاست و میتوان به آن دسترسی داشت.

برای اینکه ببینیم همه پادها رکوست میخورد از ایمج بخش اول تمرین به عنوان یک پاد درخواست دهنده استفاده میکنیم.