

گزارش تمرین دوم رایانش ابری

بخش اول

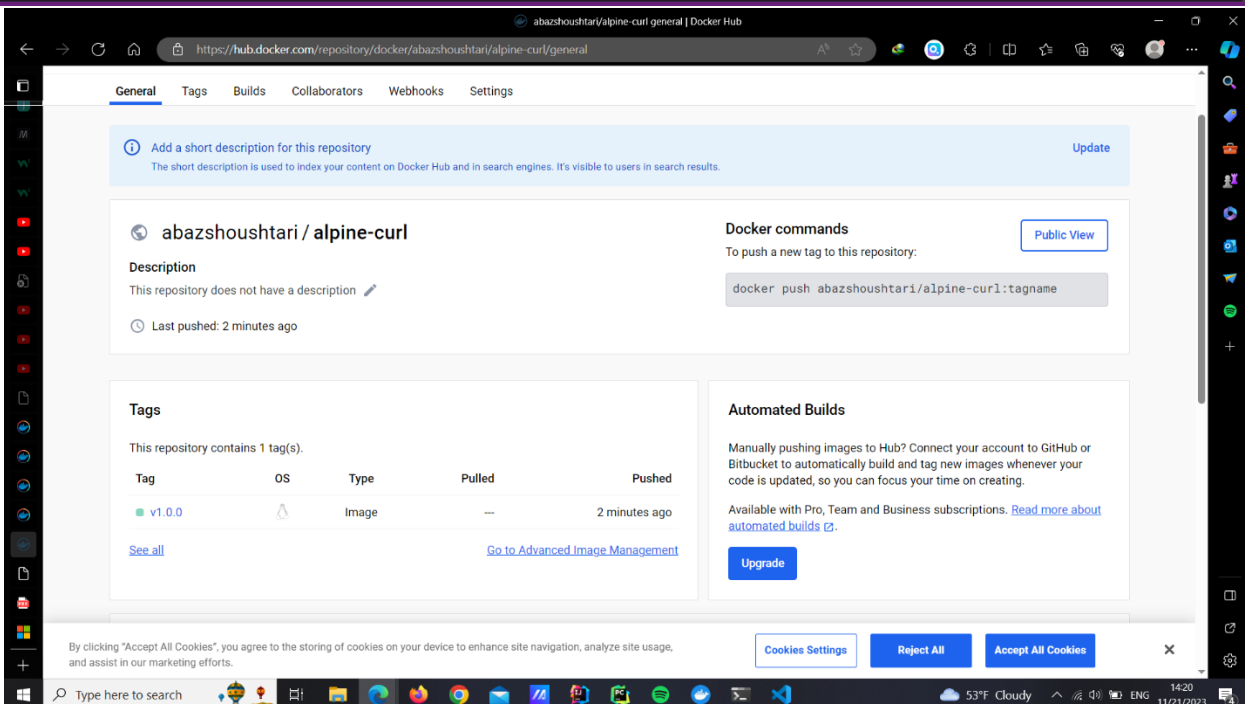
گام اول

- ارسال ایمج ساخته شده بر روی داکرهاب

```
ali@LAPTOP-6SN00ADK:~/CloudComputing/HW2_p1$ docker build --tag alpine-curl:v1.0.0 .
[+] Building 54.2s (7/7) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 87B
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load metadata for docker.io/library/alpine:latest
=> [auth] library/alpine:pull token for registry-1.docker.io
=> [1/2] FROM docker.io/library/alpine:latest@sha256:eece025e432126ce23f223450a0326fbebde39cdf496a85d8c016293fc851978
=> => resolve docker.io/library/alpine:latest@sha256:eece025e432126ce23f223450a0326fbebde39cdf496a85d8c016293fc851978
=> => sha256:48d9183eb12a05c99bcc0bf44a003607b8e941e1d4f1f9ad12bdcc4b5672f86 528B / 528B
=> => sha256:8ca4688f4f356596b5ae539337c9941abc78eda10021d35cbc52659c74d9b443 1.47kB / 1.47kB
=> => sha256:96526aa774ef0126ad0fe9e9a95764c5fc37f409ab9e97021e7b4775d82bf6fa 3.40MB / 3.40MB
```

```
PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE
=> => exporting layers
=> => writing image sha256:c694942dc50914057a8ae1c8bd384cae9f0044635bed5b40ca12b6b40687374
=> => naming to docker.io/library/alpine-curl:v1.0.0

What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview
ali@LAPTOP-6SN00ADK:~/CloudComputing/HW2_p1$ docker tag alpine-curl:v1.0.0 abazshoushtari/alpine-curl:v1.0.0
ali@LAPTOP-6SN00ADK:~/CloudComputing/HW2_p1$ docker push abazshoushtari/alpine-curl:v1.0.0
The push refers to repository [docker.io/abazshoushtari/alpine-curl]
1638cf45fd80: Pushed
cc2447e1835a: Mounted from library/alpine
v1.0.0: digest: sha256:52e2aa60605e9f199c10296bddbaf18c7d8980af2a7e363bde7f6f9f65b43fb9 size: 738
ali@LAPTOP-6SN00ADK:~/CloudComputing/HW2_p1$
```



- نمایش لیست ایمج های موجود بر روی سیستم خود
- دریافت ایمج ساخته شده از داکرهاب
- ساختن کانتینر از ایمج دریافت شده از داکرهاب
- اجرا دستور curl

The screenshot shows a Docker terminal window with the following commands and output:

```

a1@LAPTOP-6SMBADK:~/CloudComputing/H02_p1$ docker tag alpine-curl:v1.0.0 abazshoustari/alpine-curl:v1.0.0
a1@LAPTOP-6SMBADK:~/CloudComputing/H02_p1$ docker push abazshoustari/alpine-curl:v1.0.0
The push refers to repository [docker.io/abazshoustari/alpine-curl]
1638cf45f80a: Pushed
cc244e1835a: Mounted from library/alpine
v1.0.0: digest: sha256:52e2aa60605e9f19c18296bdfb18c7d8980af2a7e36bde7f69f65b43fb9 size: 738
a1@LAPTOP-6SMBADK:~/CloudComputing/H02_p1$ docker run -it abazshoustari/alpine-curl:v1.0.0
/ # curl --help
Usage: curl [options...] curl
  -d, --data <data>          HTTP POST data
  -f, --fail                  Fail fast with no output on HTTP errors
  -h, --help <category>     Get help for commands
  -i, --include                Include protocol response headers in the output
  -o, --output <file>        Write to file instead of stdout
  -O, --remote-name            Write output to a file named as the remote file
  -s, --silent                 Silent mode
  -T, --upload-file <file>    Transfer local FILE to destination
  -u, --user <user:password>  Server user and password
  -A, --user-agent <name>     Send User-Agent <name> to server
  -v, --verbose                Make the operation more talkative
  -V, --version                Show version number and quit

This is not the full help, this menu is stripped into categories.
Use "--help category" to get an overview of all categories.
For all options use the manual or "--help all".
/ # 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>
/ #

```

Below the terminal window, the Docker Desktop interface shows the following information:

- CONTAINERS:** Individual Containers, abazshoustari/alpine-curl:v1.0.0
- IMAGES:** alpine-curl, docker.io/abazshoustari/alpine-curl
- REGISTRIES:** Connect Registry...
- NETWORKS:**
- VOLUMES:**
- CONTEXTS:**
- HELP AND FEEDBACK:** Read Extension Documentation, Watch Extension Tutorial Videos, Get Started with Docker Tutorial, Open Docker Extension Walkthrough, Review Issues

The bottom status bar shows: WSL: Ubuntu, 0 errors, 0 warnings, 0 info, 53°F Cloudy, 14:30, 11/21/2023.

Docker commit

```
87d4cda9137a  abazshoushtari/alpine-curl:v1.0.0  "sh"  6 days ago  Up About a minute  adoring_driscoll
● ali@LAPTOP-6SN00ADK:~/CloudComputing/Hw2_p1$ docker commit 87d4cda9137a abazshoushtari/alpine-curl-using-commit:v1.0.0
sha256:01f5bf20220aee4e94e7abd2948b6b5d7f615bd6c8a5ef8992c9873e4e14e044
○ ali@LAPTOP-6SN00ADK:~/CloudComputing/Hw2_p1$
```

گام دوم

- دریافت ایمج ردیس و ساختن کانتینر از آن

The screenshot shows the Docker Desktop interface with a terminal window open. The terminal displays the following commands and output:

```
ali@LAPTOP-6SN00ADK:~/CloudComputing/Hw2_p1$ cd ..
ali@LAPTOP-6SN00ADK:~/CloudComputing/Hw2_p1$ cd ..
ali@LAPTOP-6SN00ADK:~/CloudComputing/Hw2_p1$ ls
CloudComputing ZeroToMastery_Python_Practice helloWorld test.pdf
ali@LAPTOP-6SN00ADK:~/CloudComputing/Hw2_p1$ cd CloudComputing/Hw2_p1/
ali@LAPTOP-6SN00ADK:~/CloudComputing/Hw2_p1$ ls
Dockerfile weatherAPI_ninjas weatherAPI_ninjas_conda
ali@LAPTOP-6SN00ADK:~/CloudComputing/Hw2_p1$ cd weatherAPI_ninjas
ali@LAPTOP-6SN00ADK:~/CloudComputing/Hw2_p1$ cd weatherAPI_ninjas_conda/
ali@LAPTOP-6SN00ADK:~/CloudComputing/Hw2_p1$ cd weatherAPI_ninjas_conda/
ali@LAPTOP-6SN00ADK:~/CloudComputing/Hw2_p1$ ls
__pycache__ main.py test_main.http
ali@LAPTOP-6SN00ADK:~/CloudComputing/Hw2_p1$ docker run redis
Unable to find image 'redis:latest' locally
latest: Pulling from library/redis
1f7ce2fa46ab: Pull complete
3c6368589bf1: Pull complete
3911d271d7d8: Pull complete
ac88aa9d4021: Pull complete
127cd75a68a2: Pull complete
4f4fb700ef54: Pull complete
f3993c1104fc: Pull complete
Digest: sha256:3f10f8fc884547e425ec7a390808e6a4e371772b9ff09391977531a8b4e4ff1
Status: Downloaded newer image for redis:latest
1:1c 21 Nov 2023 15:36:01.438 # WARNING Memory overcommit must be enabled! Without it, a background save or replication may fail under low memory condition. Bei
ng disabled, it can also cause failures without low memory condition, see https://github.com/jemalloc/jemalloc/issues/1328. To fix this issue add 'vm.overcommi
t_memory = 1' to /etc/sysctl.conf and then reboot or run the command 'sysctl vm.overcommit_memory=1' for this to take effect.
1:1c 21 Nov 2023 15:36:01.438 # 000000000000 Redis is starting 000000000000
1:1c 21 Nov 2023 15:36:01.438 # Redis version=7.2.3, bits=64, commit=00000000, modified=0, pid=1, just started
1:1c 21 Nov 2023 15:36:01.438 # Warning: no config file specified, using the default config. In order to specify a config file use redis-server /path/to/redis.c
onf
1:1M 21 Nov 2023 15:36:01.438 * monotonic clock: POSIX clock_gettime
1:1M 21 Nov 2023 15:36:01.439 * Running mode=standalone, port=6379.
1:1M 21 Nov 2023 15:36:01.439 * Server initialized
1:1M 21 Nov 2023 15:36:01.439 * Ready to accept connections tcp
```

- ساختن شبکه برای برقراری ارتباط بین دو کانتینر
- ساختن ایمج سرور نوشته شده با استفاده از داکر فایل

```

ali@LAPTOP-6SN00ADK:~/CloudComputing/HW2_p1/weatherAPI_ninjas_conda$ docker build -t cc_hw2p1_weather_app .
[+] Building 3.4s (10/10) FINISHED
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 346B
=> [internal] load metadata for docker.io/library/python:3.10-alpine
=> [1/5] FROM docker.io/library/python:3.10-alpine@sha256:28986068e3a715708b7f2e0fc4c06c748f882fb76403424c2e854483fb45304
=> [internal] load build context
=> => transferring context: 4.22kB
=> CACHED [2/5] WORKDIR /app
=> CACHED [3/5] COPY ./requirements.txt /app/requirements.txt
=> CACHED [4/5] RUN pip install -r requirements.txt
=> [5/5] COPY . .
=> exporting to image
=> => writing image sha256:1e3a5ce56f01e76b5f25100d409702a5a51e24e8ded2dc267c5edd6f0beba25e
=> => naming to docker.io/library/cc_hw2p1_weather_app

What's Next?
View a summary of image vulnerabilities and recommendations -> docker scout quickview
ali@LAPTOP-6SN00ADK:~/CloudComputing/HW2_p1/weatherAPI_ninjas_conda$ docker network create mynetwork
d1e67a05ffd03036f92d3e03adbf5ae986c33668fff785348b3c71dd5e2841
ali@LAPTOP-6SN00ADK:~/CloudComputing/HW2_p1/weatherAPI_ninjas_conda$ docker run --name redis --network mynetwork -d redis
c4adca4a7d5a6cece1db0296c5fc3d4593b1b764bbe0d56e7807850a37d6006e
ali@LAPTOP-6SN00ADK:~/CloudComputing/HW2_p1/weatherAPI_ninjas_conda$ docker run --name cc_hw2p1_weather_app --network mynetwork -p 8000:8000 -d cc_hw2p1_weather_app
918b766652aa7a1e18a992bfc04912d73e3e5dfd5b49ffbe5c3b82a0e7d4fef3
ali@LAPTOP-6SN00ADK:~/CloudComputing/HW2_p1/weatherAPI_ninjas_conda$
  
```

FastAPI - Swagger UI

localhost:8000/docs/#/ Root

Parameters

Name	Description
city	
string (query)	tehran

Execute Clear

Responses

200

Response body

```

{"cloud_pct": 0, "temp": 14, "feels_like": 12, "humidity": 22, "min_temp": 14, "max_temp": 14, "wind_speed": 2.06, "wind_direction": 210, "sunrise": 1700795921, "sunset": 1700832203}
  
```

Response headers

curl -X 'GET' \
 'http://localhost:8000/?city=tehran' \
 -H 'accept: application/json'

Request URL

http://localhost:8000/?city=tehran

Server response

Code Details

200

Response body

```

{"cloud_pct": 0, "temp": 14, "feels_like": 12, "humidity": 22, "min_temp": 14, "max_temp": 14, "wind_speed": 2.06, "wind_direction": 210, "sunrise": 1700795921, "sunset": 1700832203}
  
```

Response headers

cc_hw2p1_weather_app

STATUS

Running (1 minute ago)

918b766652aa

8000:8000

Logs

Time	Message
2023-11-24 16:27:58	INFO: Started server process [1]
2023-11-24 16:27:58	INFO: Waiting for application startup.
2023-11-24 16:27:58	INFO: Application startup complete.
2023-11-24 16:27:58	INFO: Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
2023-11-24 16:28:29	INFO: 172.19.0.1:38548 - "GET /docs HTTP/1.1" 200 OK
2023-11-24 16:28:29	INFO: 172.19.0.1:38548 - "GET /openapi.json HTTP/1.1" 200 OK
2023-11-24 16:28:39	INFO: ["cloud_pct": 0, "temp": 14, "feels_like": 12, "humidity": 22, "min_temp": 14, "max_temp": 14, "wind_speed": 2.06, "wind_direction": 210, "sunrise": 1700795921, "sunset": 1700832203]
2023-11-24 16:28:39	INFO: 172.19.0.1:38564 - "GET /?city=tehran HTTP/1.1" 200 OK
2023-11-24 16:29:17	INFO: Reading tehran response from cache
2023-11-24 16:29:17	INFO: 172.19.0.1:38976 - "GET /?city=tehran HTTP/1.1" 200 OK

• ساختن Volume جهت persist کردن اطلاعات کش ردیس

```
ali@LAPTOP-6SN00ADK:~/CloudComputing/Hw2_p1/weatherAPI_ninjas_conda$ docker volume create redis_data
redis_data
ali@LAPTOP-6SN00ADK:~/CloudComputing/Hw2_p1/weatherAPI_ninjas_conda$ docker run --name redis --network mynetwork -v redis_data:/data -d redis
bb021967a56123482e275861e39111c33f4246a0f8f42e1b4f45db5c1defc295

ali@LAPTOP-6SN00ADK:~/CloudComputing/Hw2_p1/weatherAPI_ninjas_conda$ docker run --name cc_hw2p1_weather_app --network mynetwork -p 8000:8000 -d cc_hw2p1_weather_app:v1.1
3e0eed88f198713e4f9595d337c25cf3e9757a85446d70ac0c1aed7c30844ce2
```

یک درخواست به سرور میزنیم و چون اطلاعات در سرور موجود نیست، API Call انجام می شود. در دفعه دوم چون اطلاعات در cache ذخیره شده است، اطلاعات از cache خوانده می شوند.

The image shows two side-by-side windows. The left window is the Swagger UI for the weather API, displaying a GET request to `/city?city=tehran` with a 200 status and a JSON response containing weather data. The right window is the Docker Desktop interface, showing the logs for the `cc_hw2p1_weather_app` container. The logs include the same GET request and response, confirming that the application is running and responding correctly.

The screenshot shows two windows. On the left is the FastAPI Swagger UI for a weather API. The 'city' query parameter is set to 'tehran'. The 'Execute' button has been clicked, and the response is displayed as a JSON object with weather data for Tehran. On the right is the Docker Desktop interface showing a container named 'cc_hw2p1_weather_app'. The logs show the application starting successfully and listening on port 8000. The status indicates it is running (41 seconds ago).

اکنون کانتینر مربوط به ردیس را stop و remove کرده و پس از آن مجدد یک کانتینر ردیس می سازیم و مجدد درخواست میزنیم و مشاهده میکنیم که مجدد اطلاعات از ردیس خوانده می شوند.

```

ali@LAPTOP-6SN08ADK:~/CloudComputing/HW2_p1/weatherAPI_ninjas_conda$ docker stop redis
redis
ali@LAPTOP-6SN08ADK:~/CloudComputing/HW2_p1/weatherAPI_ninjas_conda$ docker rm redis
redis
ali@LAPTOP-6SN08ADK:~/CloudComputing/HW2_p1/weatherAPI_ninjas_conda$ docker run --name redis --network mynetwork -v redis_data:/data -d redis
6deebdc1d79eade4587daade6772db6ed358ba92547f591dd91fb0fcdcb41b6
    
```

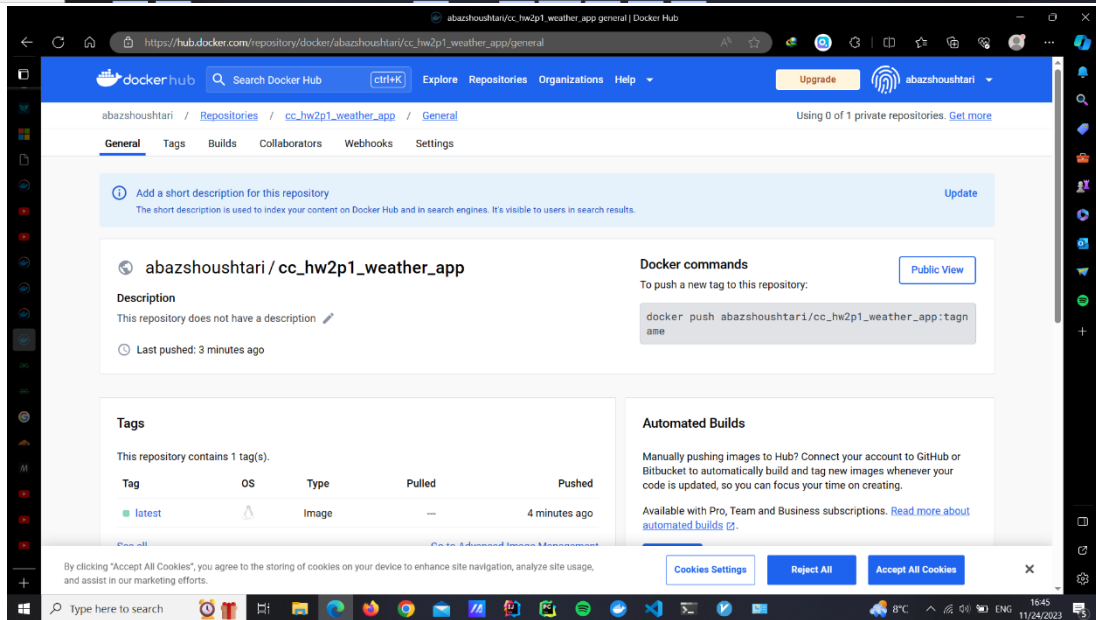
This screenshot is similar to the first one, but the Docker Desktop window shows the container 'cc_hw2p1_weather_app' has been running for 4 minutes. The logs show the application starting and listening on port 8000. The Swagger UI on the left shows the same 'tehran' query and the same JSON response, indicating that the application is still using the data from the Redis database, which was restarted in the previous step.

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

```

cc2447e1835a: Waiting
denied: requested access to the resource is denied
❯ alil@LAPTOP-6SN08QADK:~/CloudComputing/Hw2_p1/weatherAPI_ninjas_conda$ docker login -u abazshoustari
Password:
Login Succeeded
❯ alil@LAPTOP-6SN08QADK:~/CloudComputing/Hw2_p1/weatherAPI_ninjas_conda$ docker push abazshoustari/cc_hw2p1_weather_app
Using default tag: latest
The push refers to repository [docker.io/abazshoustari/cc_hw2p1_weather_app]
e7d1a178523d: Pushing [=====] 18.94kB
f69272a53969: Pushing [=====] 30.11MB
1fbf5f4f9f65: Pushing [=====] 2.56kB
8ca537607768: Retrying in 1 second
bd5ef547d2c5: Pushing [=====] 11.60MB
351cad859054: Waiting
655198633c3b: Waiting
6f25d7d19389: Waiting
cc2447e1835a: Waiting
net/http: TLS handshake timeout
❯ alil@LAPTOP-6SN08QADK:~/CloudComputing/Hw2_p1/weatherAPI_ninjas_conda$ docker push abazshoustari/cc_hw2p1_weather_app
Using default tag: latest
The push refers to repository [docker.io/abazshoustari/cc_hw2p1_weather_app]
e7d1a178523d: Retrying in 1 second
f69272a53969: Pushing [=====] 30.11MB
1fbf5f4f9f65: Retrying in 1 second
8ca537607768: Retrying in 1 second
bd5ef547d2c5: Layer already exists
351cad859054: Retrying in 14 seconds
655198633c3b: Waiting
6f25d7d19389: Waiting
cc2447e1835a: Waiting
net/http: TLS handshake timeout
❯ alil@LAPTOP-6SN08QADK:~/CloudComputing/Hw2_p1/weatherAPI_ninjas_conda$ docker push abazshoustari/cc_hw2p1_weather_app
Using default tag: latest
The push refers to repository [docker.io/abazshoustari/cc_hw2p1_weather_app]
e7d1a178523d: Pushed
f69272a53969: Layer already exists
1fbf5f4f9f65: Pushed
8ca537607768: Pushed
bd5ef547d2c5: Layer already exists
351cad859054: Pushed
655198633c3b: Layer already exists
6f25d7d19389: Layer already exists
cc2447e1835a: Layer already exists
latest: digest: sha256:ed869c5c8248254e15aa8b857c9c87dcfca5643c5f3f7e9535acdb40908ae83 size: 2201
❯ alil@LAPTOP-6SN08QADK:~/CloudComputing/Hw2_p1/weatherAPI_ninjas_conda$

```



- نمایش اطلاعات ایمیج سرور خود با استفاده از دستور `docker inspect`

```

ali@LAPTOP-6SN00ADK:~/CloudComputing/HW2_p1/weatherAPI_ninjas_conda$ docker inspect abazshoushtari/cc_hw2p1_weather_app:latest
[
  {
    "Id": "sha256:1e3a5ce56f01e7b5f25100d09702a5d51e24e8ded2c267c5edd6f0beba25e",
    "RepoTags": [
      "abazshoushtari/cc_hw2p1_weather_app:latest",
      "cc_hw2p1_weather_app:latest"
    ],
    "RepoDigests": [
      "abazshoushtari/cc_hw2p1_weather_app@sha256:ed069c5c8248254e15aa8b857c9c07dcfaa5643c5f3f7e9535acdb40908ae83"
    ],
    "Parent": "",
    "Comment": "buildkit.dockerfile.v0",
    "Created": "2023-11-24T12:54:11.684288632Z",
    "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,
    }
  }
]

```

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

```

ali@LAPTOP-6SN00ADK:~/CloudComputing/HW2_p1/weatherAPI_ninjas_conda$ docker ps

```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
918b76652aa	cc_hw2p1_weather_app	"uvicorn main:app --..."	30 minutes ago	Up 30 minutes	0.0.0.0:8000->8000/tcp	cc_hw2p1_weather_app
c4adca4a7d5a	redis	"docker-entrypoint.s..."	31 minutes ago	Up 31 minutes	6379/tcp	redis

- نمایش میزان منابع استفاده شده توسط کانتینرهای موجود با استفاده از دستور `docker stats`

CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %	NET I/O	BLOCK I/O	PIDS
742ddb1ac1d7	cc_hw2p1_weather_app	0.37%	40.05MiB / 11.58GiB	0.34%	13.6kB / 7.41kB	0B / 0B	2
ad90578c2f56	redis	0.27%	10.07MiB / 11.58GiB	0.08%	3.05kB / 939B	0B / 0B	6

بخش دوم

گزارش بخش ۱

A.

```
ali@LAPTOP-6SN00ADK:~$ minikube status
```

```
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured
```

```
ali@LAPTOP-6SN00ADK:~$ kubectl get node
```

NAME	STATUS	ROLES	AGE	VERSION
minikube	Ready	control-plane	26h	v1.28.3

گزارش بخش ۲ و ۳

2A.

در این قسمت فایل های yml. مربوطه برای Deployment، Service و ConfigMap را مینویسم.

همانطور که در پایین مشاهده می شود دیپلویمنت، پادهای مورد نیاز را ساخته است.

```
● ali@LAPTOP-6SN00ADK:~/CloudComputing/Hw2_p2/weatherAPI_minikube$ kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
redis-54d86b576b-bhqqb	1/1	Running	0	2m23s
weatherapp-deployment-cc7c44cd7-667mb	1/1	Running	0	2m23s
weatherapp-deployment-cc7c44cd7-7xlr5	1/1	Running	0	2m23s
weatherapp-deployment-cc7c44cd7-b47b9	1/1	Running	0	2m23s

نام پادها بر اساس نامی که در دیپلویمنت مشخص کرده ایم مشخص می شود و در آخر آنها یک hash اضافه می شود.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: weatherapp-deployment
  labels:
    app: weatherapp
```

2B

در این قسمت IP پادها را مشاهده میکنیم. سپس اقدام به حذف یکی از پادها میکنیم و مشاهده میکنیم که IP پاد عوض می شود.

```
ali@LAPTOP-6SN00ADK:~/CloudComputing/HW2_p2/weatherAPI_minikube$ kubectl get pods -o wide
NAME                                READY  STATUS   RESTARTS  AGE  IP            NODE      NOMINATED NODE  READINESS GATES
redis-54d86b576b-bhqqb              1/1    Running  0          2m46s  10.244.0.3    minikube  <none>          <none>
weatherapp-deployment-cc7c44cd7-667mb 1/1    Running  0          2m46s  10.244.0.6    minikube  <none>          <none>
weatherapp-deployment-cc7c44cd7-7xlr5 1/1    Running  0          2m46s  10.244.0.4    minikube  <none>          <none>
weatherapp-deployment-cc7c44cd7-b47b9 1/1    Running  0          2m46s  10.244.0.5    minikube  <none>          <none>
ali@LAPTOP-6SN00ADK:~/CloudComputing/HW2_p2/weatherAPI_minikube$ kubectl delete weatherapp-deployment-cc7c44cd7-667mb
error: the server doesn't have a resource type "weatherapp-deployment-cc7c44cd7-667mb"
ali@LAPTOP-6SN00ADK:~/CloudComputing/HW2_p2/weatherAPI_minikube$ kubectl delete pod weatherapp-deployment-cc7c44cd7-667mb
pod "weatherapp-deployment-cc7c44cd7-667mb" deleted
ali@LAPTOP-6SN00ADK:~/CloudComputing/HW2_p2/weatherAPI_minikube$ kubectl get pods -o wide
NAME                                READY  STATUS   RESTARTS  AGE  IP            NODE      NOMINATED NODE  READINESS GATES
redis-54d86b576b-bhqqb              1/1    Running  0          5m7s  10.244.0.3    minikube  <none>          <none>
weatherapp-deployment-cc7c44cd7-7xlr5 1/1    Running  0          5m7s  10.244.0.4    minikube  <none>          <none>
weatherapp-deployment-cc7c44cd7-b47b9 1/1    Running  0          5m7s  10.244.0.5    minikube  <none>          <none>
weatherapp-deployment-cc7c44cd7-x5tc6 1/1    Running  0          9s    10.244.0.7    minikube  <none>          <none>
```

در نتیجه از آیی نمیتوان استفاده کرد و باید سرویس تعریف کرد تا آیی ثابتی وجود داشته باشد.

2C

سرویس را به صورت cluster ip تعریف می کنیم که خود سرویس یک آیی ثابت می گیرد. هر موقع که نیاز به درخواست زدن به یک پاد داشته باشیم، به این سرویس و آیی درخواست می زنیم و سرویس درخواست ما را به یکی از پادها می فرستد. این نوع فقط از داخل کلاستر قابل دستیابی است و از بیرون کلاستر نمی توان به آن درخواست زد. برای درخواست های خارج از کلاستر از node port استفاده می کنیم.

دلیل اینکه نیاز نیست نوع سرویس را حتما نوع سرویس را مشخص کنیم آن است که سرویس هایی که تعریف می کنیم به صورت پیش فرض cluster ip هستند.

2D

```
ali@LAPTOP-6SN00ADK:~/CloudComputing/HW2_p2/weatherAPI_minikube$ kubectl get svc/weatherapp-service -oyaml
apiVersion: v1
kind: Service
metadata:
  annotations:
    kubectl.kubernetes.io/last-applied-configuration: |
      {"apiVersion":"v1","kind":"Service","metadata":{"annotations":{},"name":"weatherapp-service","namespace":"default"},"spec":{"ports":[{"port":8000,"protocol":"TCP","targetPort":8000}],"selector":{"app":"weatherapp"}}}
  creationTimestamp: "2023-11-26T11:01:01Z"
  name: weatherapp-service
  namespace: default
  resourceVersion: "1855"
  uid: 4225786e-8ad7-4599-9457-580f94c45314
spec:
  clusterIP: 10.97.61.60
  clusterIPs:
  - 10.97.61.60
  internalTrafficPolicy: Cluster
  ipFamilies:
  - IPv4
  ipFamilyPolicy: SingleStack
  ports:
  - port: 8000
    protocol: TCP
    targetPort: 8000
  selector:
    app: weatherapp
  sessionAffinity: None
  type: ClusterIP
status:
  loadBalancer: {}
```

3

ابتداءً ردیس را پاک میکنیم. پس از مدتی دیپلویمنت یک پاد جدید برای ردیس می سازد.

```
ali@LAPTOP-6SN00ADK:~/CloudComputing/HW2_p2/weatherAPI_minikube$ kubectl delete pod redis-54d86b576b-6mpng
pod "redis-54d86b576b-6mpng" deleted
ali@LAPTOP-6SN00ADK:~/CloudComputing/HW2_p2/weatherAPI_minikube$ k get pods
NAME                                READY   STATUS    RESTARTS   AGE
redis-54d86b576b-98j4c              1/1     Running   0           13s
weatherapp-deployment-cc7c44cd7-4jzt5 1/1     Running   0           44m
weatherapp-deployment-cc7c44cd7-8nqs9 1/1     Running   0           44m
weatherapp-deployment-cc7c44cd7-hrg5q 1/1     Running   0           44m
```

پس از دوباره زدن دستور curl مشاهده می کنیم که کش پاک شده و مجدد API Call انجام می شود.

```
/ # curl http://weatherapp-service:8000
{"cloud_pct": 75, "temp": 15, "feels like": 13, "humidity": 38, "min_temp": 10, "max_temp": 15, "wind_speed": 2.06, "wind_degrees": 60, "sunrise": 1701055295, "sunset": 1701091340}\nhostname: 10.244.0.74"/ #
```

```

● ali@LAPTOP-6SN00ADK:~/CloudComputing/HW2_p2/weatherAPI_minikube$ k logs weatherapp-deployment-bc5b8db8d-9gm6q
INFO:      Started server process [1]
INFO:      Waiting for application startup.
INFO:      Application startup complete.
INFO:      Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
{"cloud_pct": 75, "temp": 15, "feels_like": 13, "humidity": 38, "min_temp": 10, "max_temp": 15, "wind_speed": 0, "wind_degrees": 0, "sunrise": 1701055295, "sunset": 1701091340}
INFO:      10.244.0.74:49016 - "GET / HTTP/1.1" 200 OK
Reading tehran response from cache. hostname: 10.244.0.74
INFO:      10.244.0.74:42914 - "GET / HTTP/1.1" 200 OK
Reading tehran response from cache. hostname: 10.244.0.74
INFO:      10.244.0.74:42918 - "GET / HTTP/1.1" 200 OK
Reading tehran response from cache. hostname: 10.244.0.74
INFO:      10.244.0.74:42924 - "GET / HTTP/1.1" 200 OK
Reading tehran response from cache. hostname: 10.244.0.74
INFO:      10.244.0.74:48550 - "GET / HTTP/1.1" 200 OK
Reading tehran response from cache. hostname: 10.244.0.74
INFO:      10.244.0.74:48564 - "GET / HTTP/1.1" 200 OK
Reading tehran response from cache. hostname: 10.244.0.74
INFO:      10.244.0.74:51000 - "GET / HTTP/1.1" 200 OK
Reading tehran response from cache. hostname: 10.244.0.74
INFO:      10.244.0.74:51016 - "GET / HTTP/1.1" 200 OK
{"cloud_pct": 75, "temp": 15, "feels_like": 13, "humidity": 38, "min_temp": 10, "max_temp": 15, "wind_speed": 2.06, "wind_degrees": 60, "sunrise": 1701055295, "sunset": 1701091340}
INFO:      10.244.0.74:38222 - "GET / HTTP/1.1" 200 OK

```

گزارش بخش ۴

A.

با استفاده از دستور `minikube addons list` می توان مشاهده نمود که `storage-provisioner` فعال است پس در نتیجه در `minikube`، `dynamic provisioning` به صورت پیش فرض فعال است.

```
ali@LAPTOP-6SN00ADK:~/CloudComputing/Hw2_p2/weatherAPI_minikube$ minikube addons list
```

ADDON NAME	PROFILE	STATUS	MAINTAINER
ambassador	minikube	disabled	3rd party (Ambassador)
auto-pause	minikube	disabled	minikube
cloud-spanner	minikube	disabled	Google
csi-hostpath-driver	minikube	disabled	Kubernetes
dashboard	minikube	disabled	Kubernetes
default-storageclass	minikube	enabled	Kubernetes
efk	minikube	disabled	3rd party (Elastic)
freshpod	minikube	disabled	Google
gcp-auth	minikube	disabled	Google
gvisor	minikube	disabled	minikube
headlamp	minikube	disabled	3rd party (kinvolk.io)
helm-tiller	minikube	disabled	3rd party (Helm)
inacel	minikube	disabled	3rd party (InAcce)
ingress	minikube	disabled	Kubernetes
ingress-dns	minikube	disabled	minikube
inspektor-gadget	minikube	disabled	3rd party (inspektor-gadget.io)
istio	minikube	disabled	3rd party (Istio)
istio-provisioner	minikube	disabled	3rd party (Istio)
kong	minikube	disabled	3rd party (Kong HQ)
kubeflow	minikube	disabled	3rd party
kubevirt	minikube	disabled	3rd party (KubeVirt)
logviewer	minikube	disabled	3rd party (unknown)
metallb	minikube	disabled	3rd party (MetalLB)
metrics-server	minikube	disabled	Kubernetes
nvidia-device-plugin	minikube	disabled	3rd party (NVIDIA)
nvidia-driver-installer	minikube	disabled	3rd party (Nvidia)
nvidia-gpu-device-plugin	minikube	disabled	3rd party (Nvidia)
olm	minikube	disabled	3rd party (Operator Framework)
pod-security-policy	minikube	disabled	3rd party (unknown)
portainer	minikube	disabled	3rd party (Portainer.io)
registry	minikube	disabled	minikube
registry-aliases	minikube	disabled	3rd party (unknown)
registry-creds	minikube	disabled	3rd party (UPMC Enterprises)
storage-provisioner	minikube	enabled	minikube
storage-provisioner-gluster	minikube	disabled	3rd party (Gluster)
storage-provisioner-rancher	minikube	disabled	3rd party (Rancher)
volumesnapshots	minikube	disabled	Kubernetes

با استفاده از دستور `kubectl get pv` می توان وضعیت `volume` و همچنین به کمک `kubectl get pvc` وضعیت `claim` را مشاهده نمود.

```
ali@LAPTOP-6SN00ADK:~/CloudComputing/Hw2_p2/weatherAPI_minikube$ kubectl get pv
```

NAME	CAPACITY	ACCESS MODES	RECLAIM POLICY	STATUS	CLAIM	STORAGECLASS	REASON	AGE
redis-volume	512Mi	RWO	Retain	Bound	default/redis-pv-claim	manual		2m37s

```
ali@LAPTOP-6SN00ADK:~/CloudComputing/Hw2_p2/weatherAPI_minikube$ kubectl get pvc
```

NAME	STATUS	VOLUME	CAPACITY	ACCESS MODES	STORAGECLASS	AGE
redis-pv-claim	Bound	redis-volume	512Mi	RWO	manual	25h

B.

خیر بعد از اختصاص یک `pv` به یک `pvc`، `pvc`های دیگر نمی توانند به آن `pv` متصل شوند. در صورتی که تمام `pvc`ها روی یک نود باشند میتوانند به یک `pv` در همان نود باند شوند.

مزیت این سیاست آن است که دیگر پادها به دیتای یکدیگر دسترسی ندارند و کوپرنیتیز دیگر نگران مدیریت این فضا نمی باشد و پیاده سازی راحت تر است.

گزارش بخش ۵

صحت اجرای سرویس و لود بالانسینگ

```
Reading tehran response from cache
INFO: 10.244.0.1:35503 - "GET / HTTP/1.1" 200 OK
Reading tehran response from cache
INFO: 10.244.0.1:31927 - "GET / HTTP/1.1" 200 OK
Reading tehran response from cache
INFO: 10.244.0.1:55282 - "GET / HTTP/1.1" 200 OK
Reading tehran response from cache
INFO: 10.244.0.1:20834 - "GET / HTTP/1.1" 200 OK
Reading tehran response from cache
INFO: 10.244.0.1:3841 - "GET / HTTP/1.1" 200 OK
Reading tehran response from cache
INFO: 10.244.0.1:24958 - "GET / HTTP/1.1" 200 OK
Reading tehran response from cache
INFO: 10.244.0.1:7372 - "GET / HTTP/1.1" 200 OK
Reading tehran response from cache
INFO: 10.244.0.1:1991 - "GET / HTTP/1.1" 200 OK
Reading tehran response from cache
INFO: 10.244.0.1:24833 - "GET / HTTP/1.1" 200 OK
Reading tehran response from cache
INFO: 10.244.0.1:53469 - "GET / HTTP/1.1" 200 OK
```

```
http://127.0.0.1:33207
{"cloud_pct": 20, "temp": 15, "feels_like": 13, "humidity": 22, "min temp": 15, "max temp": 15, "wind speed": 2.06, "wind degrees": 150, "sunrise": 1700968837, "sunset": 1701004959}
ali@LAPTOP-6SN00ADK:~/CloudComputing/HW2_p2/weatherAPI_minikube$ curl http://127.0.0.1:33207
{"cloud_pct": 20, "temp": 15, "feels_like": 13, "humidity": 22, "min temp": 15, "max temp": 15, "wind speed": 2.06, "wind degrees": 150, "sunrise": 1700968837, "sunset": 1701004959}
ali@LAPTOP-6SN00ADK:~/CloudComputing/HW2_p2/weatherAPI_minikube$ curl http://127.0.0.1:33207
{"cloud_pct": 20, "temp": 15, "feels_like": 13, "humidity": 22, "min temp": 15, "max temp": 15, "wind speed": 2.06, "wind degrees": 150, "sunrise": 1700968837, "sunset": 1701004959}
ali@LAPTOP-6SN00ADK:~/CloudComputing/HW2_p2/weatherAPI_minikube$ curl http://127.0.0.1:33789
^[[A{"cloud_pct": 20, "temp": 14, "feels_like": 12, "humidity": 26, "min temp": 12, "max temp": 14, "wind speed": 0, "wind degrees": 0, "sunrise": 1700968837, "sunset": 1701004959}
ali@LAPTOP-6SN00ADK:~/CloudComputing/HW2_p2/weatherAPI_minikube$ curl http://127.0.0.1:33789
```

<pre>ali@LAPTOP-6SN00ADK:~/CloudComputing/HW2_p2/weatherAPI_minikube\$ k logs -f weatherapp-deployment-cc7c44cd7-hrg5q INFO: Started server process [1] INFO: Waiting for application startup. INFO: Application startup complete. INFO: Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)</pre>	<pre>ali@LAPTOP-6SN00ADK:~/CloudComputing/HW2_p2/weatherAPI_minikube\$ k logs -f weatherapp-deployment-cc7c44cd7-hrg5q INFO: Started server process [1] INFO: Waiting for application startup. INFO: Application startup complete. INFO: Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit) Reading tehran response from cache INFO: 10.244.0.1:5399 - "GET / HTTP/1.1" 200 OK</pre>	<pre>ali@LAPTOP-6SN00ADK:~/CloudComputing/HW2_p2/weatherAPI_minikube\$ k logs -f weatherapp-deployment-cc7c44cd7-hrg5q INFO: Started server process [1] INFO: Waiting for application startup. INFO: Application startup complete. INFO: Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)</pre>	<pre>e\$ curl http://127.0.0.1:35109 {"cloud_pct": 20, "temp": 14, "feels_like": 12, "humidity": 26, "min temp": 12, "max temp": 14, "wind speed": 0, "wind degrees": 0, "sunrise": 1700968837, "sunset": 1701004959} ali@LAPTOP-6SN00ADK:~/CloudComputing/HW2_p2/weatherAPI_minikube\$ curl http://127.0.0.1:35109 {"cloud_pct": 20, "temp": 14, "feels_like": 12, "humidity": 26, "min temp": 12, "max temp": 14, "wind speed": 0, "wind degrees": 0, "sunrise": 1700968837, "sunset": 1701004959} ali@LAPTOP-6SN00ADK:~/CloudComputing/HW2_p2/weatherAPI_minikube\$</pre>	<pre>minikube w... r kubectl w... k kubectl w... bash weat...</pre>
--	---	--	--	---

Load balancing که چند تا دستور curl زده شده اما همگی به یک پاد نخورده و پس از چند درخواست به یکی از پادها خورده

```
ali@LAPTOP-6SN00ADK:~/CloudComputing/HW2_p2/weatherAPI_minikube$ k get pods
NAME                                READY   STATUS    RESTARTS   AGE
alpine-curl                        1/1     Running   4 (168m ago)  25h
redis-54d86b576b-j6gcp             1/1     Running   6 (168m ago)  28h
weatherapp-deployment-bc5b8db8d-9gm6q 1/1     Running   3 (168m ago)  25h
weatherapp-deployment-bc5b8db8d-fq5mz 1/1     Running   3 (168m ago)  25h
weatherapp-deployment-bc5b8db8d-hkv8d 1/1     Running   3 (168m ago)  25h
ali@LAPTOP-6SN00ADK:~/CloudComputing/HW2_p2/weatherAPI_minikube$ kubectl exec -it alpine-curl -- sh
/ # curl http://weatherapp-service:8000
{"cloud_pct": 75, "temp": 15, "feels_like": 13, "humidity": 38, "min_temp": 10, "max_temp": 15, "wind_speed": 0, "wind_degrees": 0, "sunrise": 1701055295, "sunset": 1701091340}
```

اجرای curl از طریق ایمیج alpine-curl که در پارت ۱ تمرین ساختیم

پس از چند بار دستور curl زدن به سرویس اکنون log پادها را مشاهده می کنیم و میبینیم که دستورات به پادهای مختلفی خورده اند.

```
INFO: Application startup complete.
INFO: Uvicorn running on http://0.0.0.0:8000
(Press CTRL+C to quit)
{"cloud_pct": 75, "temp": 15, "feels_like": 13, "humidity": 38, "min_temp": 10, "max_temp": 15, "wind_speed": 0, "wind_degrees": 0, "sunrise": 1701055295, "sunset": 1701091340}
INFO: 10.244.0.74:49016 - "GET / HTTP/1.1" 200 OK
ali@LAPTOP-6SN00ADK:~/CloudComputing/HW2_p2/weatherAPI_minikube$ k logs weatherapp-deployment-bc5b8db8d-9gm6q
INFO: Started server process [1]
INFO: Waiting for application startup.
INFO: Application startup complete.
INFO: Uvicorn running on http://0.0.0.0:8000
(Press CTRL+C to quit)
{"cloud_pct": 75, "temp": 15, "feels_like": 13, "humidity": 38, "min_temp": 10, "max_temp": 15, "wind_speed": 0, "wind_degrees": 0, "sunrise": 1701055295, "sunset": 1701091340}
INFO: 10.244.0.74:49016 - "GET / HTTP/1.1" 200 OK
Reading tehran response from cache. hostname: 10.244.0.74
INFO: 10.244.0.74:42914 - "GET / HTTP/1.1" 200 OK
Reading tehran response from cache. hostname: 10.244.0.74
INFO: 10.244.0.74:42918 - "GET / HTTP/1.1" 200 OK
Reading tehran response from cache. hostname: 10.244.0.74
INFO: 10.244.0.74:42924 - "GET / HTTP/1.1" 200 OK
Reading tehran response from cache. hostname: 10.244.0.74
INFO: 10.244.0.74:48550 - "GET / HTTP/1.1" 200 OK
Reading tehran response from cache. hostname: 10.244.0.74
INFO: 10.244.0.74:48564 - "GET / HTTP/1.1" 200 OK
ali@LAPTOP-6SN00ADK:~/CloudComputing/HW2_p2/weatherAPI_minikube$

ali@LAPTOP-6SN00ADK:~/CloudComputing/HW2_p2/weatherAPI_minikube$ k logs weatherapp-deployment-bc5b8db8d-fq5mz
INFO: Started server process [1]
INFO: Waiting for application startup.
INFO: Application startup complete.
INFO: Uvicorn running on http://0.0.0.0:8000
(Press CTRL+C to quit)
ali@LAPTOP-6SN00ADK:~/CloudComputing/HW2_p2/weatherAPI_minikube$ k logs weatherapp-deployment-bc5b8db8d-hkv8d
INFO: Started server process [1]
INFO: Waiting for application startup.
INFO: Application startup complete.
INFO: Uvicorn running on http://0.0.0.0:8000
(Press CTRL+C to quit)
Reading tehran response from cache. hostname: 10.244.0.74
INFO: 10.244.0.74:51020 - "GET / HTTP/1.1" 200 OK
ali@LAPTOP-6SN00ADK:~/CloudComputing/HW2_p2/weatherAPI_minikube$

ali@LAPTOP-6SN00ADK:~/CloudComputing/HW2_p2/weatherAPI_minikube$ k logs weatherapp-deployment-bc5b8db8d-hkv8d
INFO: Started server process [1]
INFO: Waiting for application startup.
INFO: Application startup complete.
INFO: Uvicorn running on http://0.0.0.0:8000
(Press CTRL+C to quit)
Reading tehran response from cache. hostname: 10.244.0.74
INFO: 10.244.0.74:42938 - "GET / HTTP/1.1" 200 OK
ali@LAPTOP-6SN00ADK:~/CloudComputing/HW2_p2/weatherAPI_minikube$ k logs weatherapp-deployment-bc5b8db8d-hkv8d
INFO: Started server process [1]
INFO: Waiting for application startup.
INFO: Application startup complete.
INFO: Uvicorn running on http://0.0.0.0:8000
(Press CTRL+C to quit)
Reading tehran response from cache. hostname: 10.244.0.74
INFO: 10.244.0.74:42938 - "GET / HTTP/1.1" 200 OK
Reading tehran response from cache. hostname: 10.244.0.74
INFO: 10.244.0.74:48544 - "GET / HTTP/1.1" 200 OK
Reading tehran response from cache. hostname: 10.244.0.74
INFO: 10.244.0.74:48556 - "GET / HTTP/1.1" 200 OK
Reading tehran response from cache. hostname: 10.244.0.74
INFO: 10.244.0.74:51348 - "GET / HTTP/1.1" 200 OK
ali@LAPTOP-6SN00ADK:~/CloudComputing/HW2_p2/weatherAPI_minikube$
```

The screenshot shows a VS Code interface with a terminal window titled "weatherAPI_minikube [WSL: Ubuntu]". The terminal displays the following commands and output:

```

ali@LAPTOP-6SN0B0ADK:~/CloudComputing/HW2_p2/weatherAPI_minikube$ k get logs
weatherapp-deployment-bc5b8db8d-9gm6q
error: the server doesn't have a resource type "logs"
ali@LAPTOP-6SN0B0ADK:~/CloudComputing/HW2_p2/weatherAPI_minikube$ k get log
weatherapp-deployment-bc5b8db8d-9gm6q
error: the server doesn't have a resource type "log"
ali@LAPTOP-6SN0B0ADK:~/CloudComputing/HW2_p2/weatherAPI_minikube$ k logs wea
therapp-deployment-bc5b8db8d-9gm6q
INFO: Started server process [1]
INFO: Waiting for application startup.
INFO: Application startup complete.
INFO: Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
ali@LAPTOP-6SN0B0ADK:~/CloudComputing/HW2_p2/weatherAPI_minikube$ k logs wea
therapp-deployment-bc5b8db8d-9gm6q
INFO: Started server process [1]
INFO: Waiting for application startup.
INFO: Application startup complete.
INFO: Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
ali@LAPTOP-6SN0B0ADK:~/CloudComputing/HW2_p2/weatherAPI_minikube$ k
logs weatherapp-deployment-bc5b8db8d-9gm6q
INFO: Started server process [1]
INFO: Waiting for application startup.
INFO: Application startup complete.
INFO: Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
ali@LAPTOP-6SN0B0ADK:~/CloudComputing/HW2_p2/weatherAPI_minikube$ k logs wea
therapp-deployment-bc5b8db8d-9gm6q
INFO: Started server process [1]
INFO: Waiting for application startup.
INFO: Application startup complete.
INFO: Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
Reading tehran response from cache. hostname: 10.244.0.54
INFO: 10.244.0.54:46050 - "GET / HTTP/1.1" 200 OK
Reading tehran response from cache. hostname: 10.244.0.54
INFO: 10.244.0.54:46052 - "GET / HTTP/1.1" 200 OK
Reading tehran response from cache. hostname: 10.244.0.54
INFO: 10.244.0.54:46064 - "GET / HTTP/1.1" 200 OK
Reading tehran response from cache. hostname: 10.244.0.54
INFO: 10.244.0.54:46076 - "GET / HTTP/1.1" 200 OK
Reading tehran response from cache. hostname: 10.244.0.54
INFO: 10.244.0.54:46080 - "GET / HTTP/1.1" 200 OK
ali@LAPTOP-6SN0B0ADK:~/CloudComputing/HW2_p2/weatherAPI_minikube$

```

The output shows the application is running successfully and returning JSON responses for weather data. The status bar at the bottom indicates the file is in WSL: Ubuntu, with 4 spaces, UTF-8 encoding, and CR/LF line endings.

The screenshot shows the VS Code editor with the following components:

- EXPLORER:** Displays the project structure for 'WEATHERAPI_MINIKUBE (WSL: UBUNTU)'. Files include 'docker-compose.yaml', 'Dockerfile', 'main.py', 'requirements.txt', 'test_main.http', 'redis-deployment.yaml', 'redis-service.yaml', 'weatherApp-configmap.yaml', 'weatherApp-deployment.yaml', and 'weatherApp-service.yaml'.
- EDITOR:** Shows the 'docker-compose.yaml' file with the following content:


```

16 container_name: redis
17 networks:
18   - mynetwork
19 volumes:
20   - redis_data:/data
21 command: ["redis-server", "--appendonly", "yes"]
22
23 networks:
24   mynetwork:
25     driver: bridge
26
27 volumes:
28   redis_data:
29

```
- TERMINAL:** Contains three terminal windows:
 - Terminal 1:** Shows the output of 'docker-compose up -d' and 'redis-cli ping', indicating the Redis container is running successfully.
 - Terminal 2:** Shows the output of 'python main.py', displaying weather data for Tehran: '{"cloud_pct": 20, "temp": 14, "humidity": 12, "max temp": 14, "min temp": 12, "wind speed": 0, "wind degrees": 0, "sunrise": 1700968837, "sunset": 1701004959}'.
 - Terminal 3:** Shows the output of 'curl http://127.0.0.1:35109', displaying the same weather data as Terminal 2.

This screenshot is identical to the one above, showing the same VS Code interface with the 'docker-compose.yaml' file and three terminal windows displaying the setup and execution of the weather API service.