## Homework 1: Microservices

https://github.com/kathyshe/Cloud-Computing/Docker

1. Build image from city-to-zip python file, then run a container from this image.

```
jaydenj@JaydendeMBP Container1 % docker build -t city-to-zip .
[+] Building 6.0s (10/10) FINISHED
 => [internal] load build definition from Dockerfile              0.0s
 => => transferring dockerfile: 311B                              0.0s
 => [internal] load .dockerignore                                 0.0s
 => => transferring context: 2B                                   0.0s
 => [internal] load metadata for docker.io/library/python:3.8    1.3s
 => [internal] load build context                                 0.0s
 => => transferring context: 1.59kB                               0.0s
 => [1/5] FROM docker.io/library/python:3.8@sha256:8b13c404b99    0.0s
 => => resolve docker.io/library/python:3.8@sha256:8b13c404b99    0.0s
 => CACHED [2/5] WORKDIR /app                                     0.0s
 => [3/5] COPY . .                                                0.0s
 => [4/5] RUN pip install Flask                                   2.9s
 => [5/5] RUN pip install requests                                1.5s
 => exporting to image                                            0.1s
 => => exporting layers                                           0.1s
 => => writing image sha256:8ec5d14fc3112d08e8a95e6861209e3bc7    0.0s
 => => naming to docker.io/library/city-to-zip                    0.0s
```

```
Use 'docker scan' to run Snyk tests against images to find vulnerabil
ities and learn how to fix them
jaydenj@JaydendeMBP Container1 % docker run --name city-zip-container
 -p 8000:8000 city-to-zip
 * Serving Flask app 'city_to_zipcode'
 * 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://127.0.0.1:8000
 * Running on http://172.17.0.2:8000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 922-605-984
```
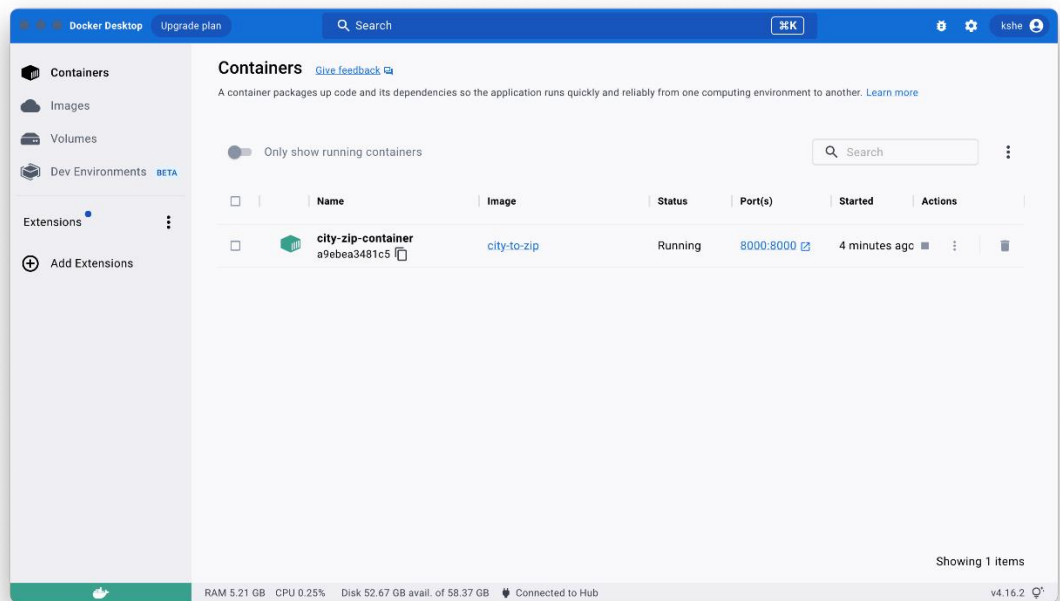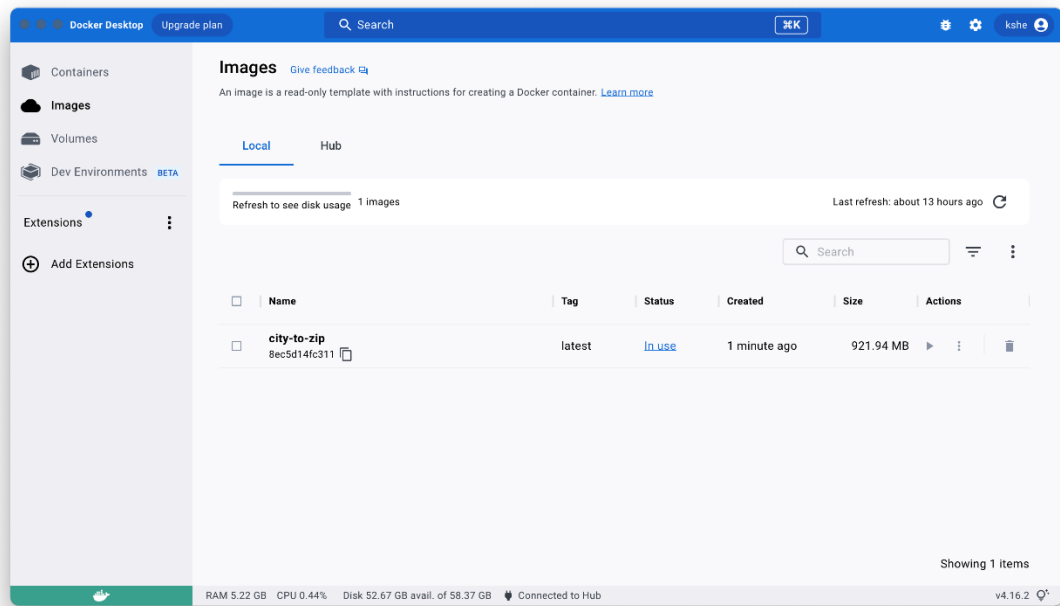
2. Build image from zip-to-weather python file, then run a container from this image.

```
jaydenj@JaydendeMBP Container2 % docker build -t zip-to-weather .
[+] Building 6.5s (11/11) FINISHED
 => [internal] load build definition from Dockerfile          0.0s
 => => transferring dockerfile: 311B                          0.0s
 => [internal] load .dockerignore                             0.0s
 => => transferring context: 2B                               0.0s
 => [internal] load metadata for docker.io/library/python:3.8 1.5s
 => [auth] library/python:pull token for registry-1.docker.io 0.0s
 => [1/5] FROM docker.io/library/python:3.8@sha256:8b13c404b99 0.0s
 => => resolve docker.io/library/python:3.8@sha256:8b13c404b99 0.0s
 => [internal] load build context                            0.0s
 => => transferring context: 7.92kB                          0.0s
 => CACHED [2/5] WORKDIR /app                                 0.0s
 => [3/5] COPY . .                                           0.0s
 => [4/5] RUN pip install requests                           2.7s
 => [5/5] RUN pip install Flask                              2.0s
 => exporting to image                                        0.1s
 => => exporting layers                                       0.1s
 => => writing image sha256:f1c41d0f3d6999b5691390aa82eb69ca73 0.0s
 => => naming to docker.io/library/zip-to-weather             0.0s
```

```
Use 'docker scan' to run Snyk tests against images to find vulnerabil
ities and learn how to fix them
jaydenj@JaydendeMBP Container2 % docker run --name zip-weather-contai
ner -p 8001:8001 zip-to-weather
 * Serving Flask app 'zipcode_to_weather'
 * 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://127.0.0.1:8001
 * Running on http://172.17.0.2:8001
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 922-605-984
```
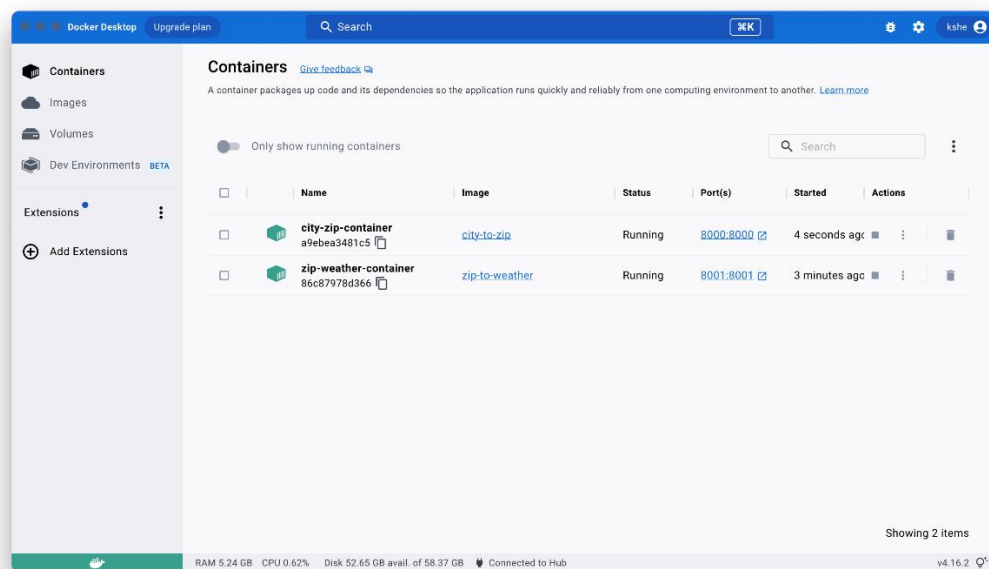
3. Create network and add containers to network

```
● jaydenj@JaydendeMBP HW1 % docker network create city—weather—network
  e2764f8898230d548864a8dbe89aa4c25c68a9dece389723be5d54e1adfc1e47
● jaydenj@JaydendeMBP HW1 % docker network ls
  NETWORK ID        NAME                          DRIVER      SCOPE
  c9d6da6b7e1d      bridge                        bridge      local
  e2764f889823      city—weather—network          bridge      local
  edf05dc968b4      host                          host        local
  b24fe04a40a4      none                          null        local
```
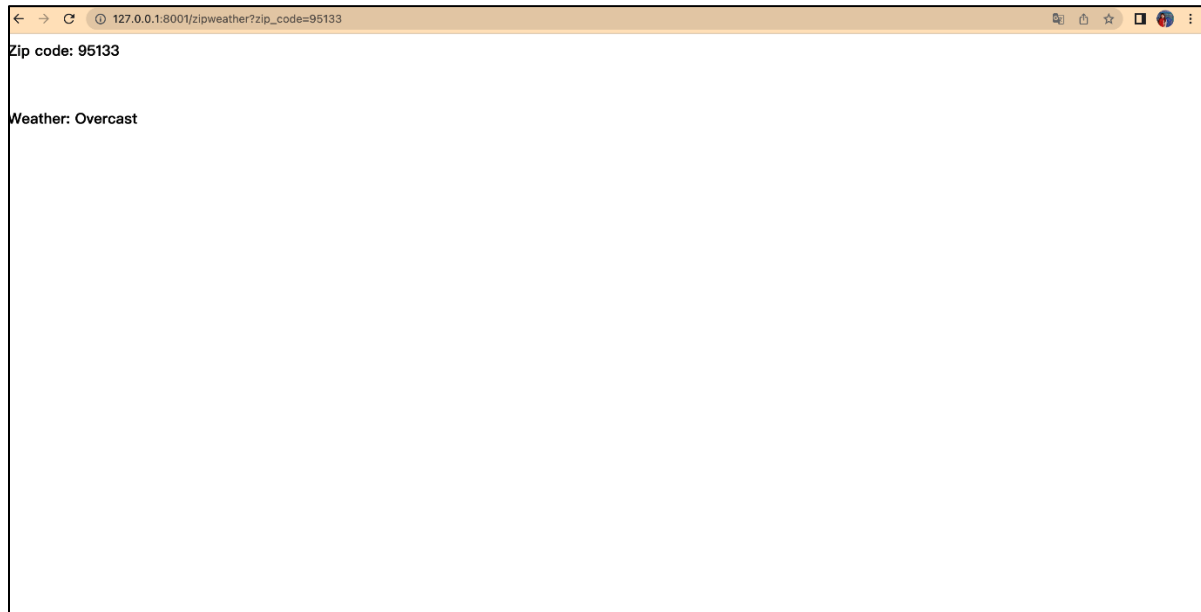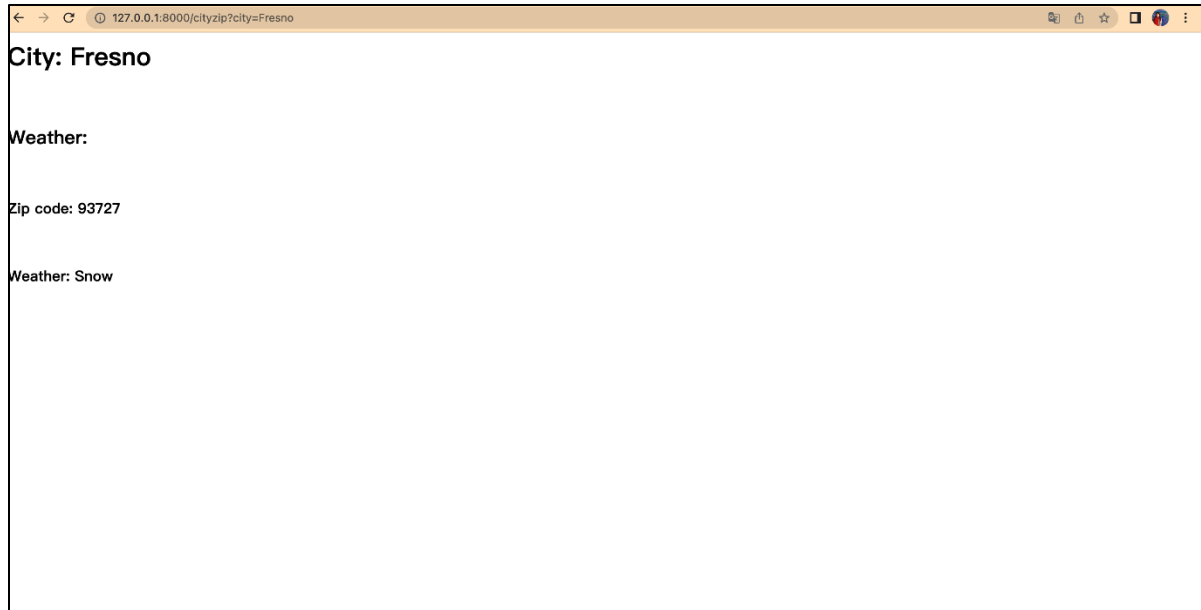
```
● jaydenj@JaydendeMBP HW1 % docker network connect city—weather—network city—zip—container
● jaydenj@JaydendeMBP HW1 % docker network connect city—weather—network zip—weather—container
● jaydenj@JaydendeMBP HW1 % docker network inspect city—weather—network
[
    {
        "Name": "city—weather—network",
        "Id": "e2764f8898230d548864a8dbe89aa4c25c68a9dece389723be5d54e1adfc1e47",
        "Created": "2023—02—09T21:13:22.027086921Z",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": {},
            "Config": [
                {
                    "Subnet": "172.18.0.0/16",
                    "Gateway": "172.18.0.1"
                }
            ]
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": {
            "86c87978d36671ef3d5a289dbdd9608123c12c521ea6a99a670209c7604db197": {
                "Name": "zip—weather—container",
                "EndpointID": "0905b7ae54c4052e0f06f34e13bcad68f7078045917211ebc2854a8e3c07659d",
                "MacAddress": "02:42:ac:12:00:03",
                "IPv4Address": "172.18.0.3/16",
                "IPv6Address": ""
            },
            "a9ebea3481c5e825b92b4f34c0242d7865d5510d54015e8f07e8f6ad09947689": {
                "Name": "city—zip—container",
                "EndpointID": "da94d42e7c6970946feaf48e07630ee9a534092216d0e3977ad3b9d7b7d036db",
                "MacAddress": "02:42:ac:12:00:02",
                "IPv4Address": "172.18.0.2/16",
                "IPv6Address": ""
            }
        },
        "Options": {},
```

4. Check via browser and curl

127.0.0.1:8000/cityzip?city=Fresno

City: Fresno

Weather:

Zip code: 93727

Weather: Snow

127.0.0.1:8001/zipweather?zip_code=95133

Zip code: 95133

Weather: Overcast

```
jaydenj@JaydendeMBP HW1 % curl "http://127.0.0.1:8000/cityzip?city=Fresno"
<h1>City: Fresno</h1><br><h2>Weather:</h2><br><h3><h3>Zip code: 93727</h3><br><h3>Weather: S
now</h3></h3>%
jaydenj@JaydendeMBP HW1 % curl "http://127.0.0.1:8001/zipweather?zip_code=10011"
<h3>Zip code: 10011</h3><br><h3>Weather: Not available</h3>%
```