

Homework 3 - Building Images

Exercise 1: Build and Containerize an API (Back-End)

Develop a simple API using a programming language of your choice (e.g., Node.js, Python, Go).

- The API must expose an endpoint (e.g., /info) that returns:
- The container's hostname
- The container's IP address

```
ubuntu@k8s-instance-6:~/backend$ ls
Dockerfile  app.py  requirement.txt  requirements.txt
ubuntu@k8s-instance-6:~/backend$ cat app.py
from flask import Flask, jsonify
from flask_cors import CORS
import socket

app = Flask(__name__)
CORS(app) # ¡habilita CORS para todas las rutas!

@app.route('/info')
def info():
    hostname = socket.gethostname()
    ip_address = socket.gethostbyname(hostname)
    return jsonify({
        'hostname': hostname,
        'ip_address': ip_address
    })

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000)
ubuntu@k8s-instance-6:~/backend$ cat requirements.txt
flask
flask-cors
ubuntu@k8s-instance-6:~/backend$ |
```

- Write a Dockerfile using **multistage build** to containerize the API.

```

ubuntu@k8s-instance-6:~/backend$ cat Dockerfile
# Stage 1: Build
FROM python:3.11-slim AS builder
WORKDIR /app
COPY requirements.txt .
RUN pip install --user -r requirements.txt

# Stage 2: Runtime
FROM python:3.11-slim
WORKDIR /app
COPY --from=builder /root/.local /root/.local
ENV PATH=/root/.local/bin:$PATH
COPY app.py .
#EXPOSE 5000
CMD ["python", "app.py"]
ubuntu@k8s-instance-6:~/backend$

```

- Build the image and run the container.

```

ubuntu@k8s-instance-6:~$ mkdir flask_docker_api
ubuntu@k8s-instance-6:~$ cd flask_docker_api/
ubuntu@k8s-instance-6:~/flask_docker_api$ vi app.py
ubuntu@k8s-instance-6:~/flask_docker_api$ vi requirements.txt
ubuntu@k8s-instance-6:~/flask_docker_api$ vi Dockerfile
ubuntu@k8s-instance-6:~/flask_docker_api$ docker build -t my-flask-api .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

Sending build context to Docker daemon  4.096kB
Step 1/11 : FROM python:3.11-slim AS builder
3.11-slim: Pulling from library/python
254e724d7786: Already exists
eb0baa05daea: Pulling fs layer
457229a5b852: Pulling fs layer
b658f584ba6e: Pulling fs layer
b658f584ba6e: Verifying Checksum
b658f584ba6e: Download complete
eb0baa05daea: Verifying Checksum
eb0baa05daea: Download complete
eb0baa05daea: Pull complete
457229a5b852: Verifying Checksum
457229a5b852: Download complete
457229a5b852: Pull complete
b658f584ba6e: Pull complete
Digest: sha256:9c85d1d49df54abca1c5db3b4016400e198e9e9bb699f32f1ef8e5c0c2149ccf
Status: Downloaded newer image for python:3.11-slim
----> bea8499a31bb
Step 2/11 : WORKDIR /app
----> Running in b87856ac4142
----> Removed intermediate container b87856ac4142
----> 04dd815ae7ff
Step 3/11 : COPY requirements.txt .
----> 8b396375a3f2
Step 4/11 : RUN pip install --upgrade pip && pip install -r requirements.txt
----> Running in 4107955ebafa
Requirement already satisfied: pip in /usr/local/lib/python3.11/site-packages (24.0)
Collecting pip
  Downloading pip-25.1.1-py3-none-any.whl.metadata (3.6 kB)
  Downloading pip-25.1.1-py3-none-any.whl (1.8 MB)
  _____ 1.8/1.8 MB 6.1 MB/s eta 0:00:00
Installing collected packages: pip

```

- Test the endpoint with curl to verify that it returns the correct information.

- Ensure the API is not exposed to the host. [The API is not exposed to the host because we did **not** use the `-p` flag when running the container.]

```
ubuntu@k8s-instance-6:~/flask_docker_api$ docker network create internal-net
bfa71fc961ec3d1c5a2306df9398cafd9908347551cdce9c43464a5842a4526
ubuntu@k8s-instance-6:~/flask_docker_api$ docker run -d --rm --name api-container --network internal-net my-flask-api
50ffaf1b42fb9878e828229796bc23359c3afd49672e58231d241ff3b92193f0
ubuntu@k8s-instance-6:~/flask_docker_api$ docker run --rm --network internal-net curlimages/curl http://api-container:5000/info
Unable to find image 'curlimages/curl:latest' locally
latest: Pulling from curlimages/curl
f18232174bc9: Pull complete
0840d8ec766c: Pull complete
bd9ddc54bea9: Pull complete
Digest: sha256:d43bdb28bae0be0998f3be83199bfb2b81e0a30b034b6d7586ce7e05de34c3fd
Status: Downloaded newer image for curlimages/curl:latest
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
 100    54    100    54    0     0    6526      0 --:--:-- --:--:-- --:--:--  7714
{"hostname":"50ffaf1b42fb","ip_address":"172.19.0.2"}
ubuntu@k8s-instance-6:~/flask_docker_api$
```

Exercise 2: Build and Containerize a Front-End Application

- Create a front-end application using HTML/JavaScript or a framework of your choice.
- The app must fetch the `/info` endpoint from the backend API and display the hostname and IP address.
- Write a Dockerfile using **multistage build** to containerize and minimize the final image.

```

ubuntu@k8s-instance-6:~/frontend$ cat Dockerfile
# Usamos nginx como servidor web estático
FROM nginx:alpine
WORKDIR /usr/share/nginx/html
COPY nginx.conf /etc/nginx/nginx.conf
COPY index.html .
CMD ["nginx", "-g", "daemon off;"]
ubuntu@k8s-instance-6:~/frontend$ cat index.html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Container Info</title>
</head>
<body>
  <h1>Container Info</h1>
  <div id="info">Loading...</div>
  <script>
    // Ahora pedimos al mismo origen: /info
    fetch('/info')
      .then(r => r.json())
      .then(data => {
        document.getElementById('info').innerHTML = `
          <p><strong>Hostname:</strong> ${data.hostname}</p>
          <p><strong>IP Address:</strong> ${data.ip_address}</p>
        `;
      })
      .catch(e => {
        document.getElementById('info').textContent = 'Failed to fetch data';
        console.error(e);
      });
  </script>
</body>
</html>
ubuntu@k8s-instance-6:~/frontend$

```

- Create a user-defined Docker network and run both frontend and backend containers within it.

```

ubuntu@k8s-instance-6:~/backend$ vi app.py
ubuntu@k8s-instance-6:~/backend$ vi requirements.txt
ubuntu@k8s-instance-6:~/backend$ vi Dockerfile
ubuntu@k8s-instance-6:~/backend$ cd ..
ubuntu@k8s-instance-6:~$ cd frontend/
ubuntu@k8s-instance-6:~/frontend$ cat nginx.conf
events { }

http {
    server {
        listen 80;

        # Sirve el HTML
        location / {
            root /usr/share/nginx/html;
            index index.html;
            try_files $uri $uri/ =404;
        }

        # Proxy para /info hacia el backend
        location /info {
            proxy_pass http://backend:5000/info;
        }
    }
}
ubuntu@k8s-instance-6:~/frontend$ cd ..
ubuntu@k8s-instance-6:~$ docker network create mynetwork
d5a462a4ff19f3231c36afe3b8b344c8a025b107d85de3767e54155db57b13f6
ubuntu@k8s-instance-6:~$ docker build -t simple-api ./backend
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
             Install the buildx component to build images with BuildKit:
             https://docs.docker.com/go/buildx/

Sending build context to Docker daemon   5.12kB
Step 1/10 : FROM python:3.11-slim AS builder
3.11-slim: Pulling from library/python

```



```

ubuntu@k8s-instance-6:~$ docker run -d \
  --name backend \
  --network mynetwork \
  simple-api
7ecc8f714f1b2453db1d58a2d6036492dee541b561b4508a5b688829f8105543
ubuntu@k8s-instance-6:~$ docker build -t frontend-image ./frontend
DEPRECATED: The legacy builder is deprecated and will be removed in a future release
Install the buildx component to build images with BuildKit:
https://docs.docker.com/go/buildx/

Sending build context to Docker daemon 4.608kB
Step 1/5 : FROM nginx:alpine
alpine: Pulling from library/nginx
Digest: sha256:65645c7bb6a0661892a8b03b89d0743208a18dd2f3f17a54ef4b76fb8e2f2a10
Status: Downloaded newer image for nginx:alpine
----> 6769dc3a703c
Step 2/5 : WORKDIR /usr/share/nginx/html
----> Running in e33d0295f8e4
----> Removed intermediate container e33d0295f8e4
----> 70534705d371
Step 3/5 : COPY nginx.conf /etc/nginx/nginx.conf
----> 801c84be4147
Step 4/5 : COPY index.html .
----> 984309d4a417
Step 5/5 : CMD ["nginx", "-g", "daemon off;"]
----> Running in 5046cdcbdb07
----> Removed intermediate container 5046cdcbdb07
----> 9466c36190a8
Successfully built 9466c36190a8
Successfully tagged frontend-image:latest
ubuntu@k8s-instance-6:~$ docker run -d \
  --name frontend \
  --network mynetwork \
  -p 8080:80 \
  frontend-image
f3b5cb5a03c68fe2fed7c0b6ff7d00dd5e956a400a4b3e7a688a100b5bef613d
ubuntu@k8s-instance-6:~$

```

- Verify in the browser that the frontend correctly shows the container metadata served by the Back End.

```

ubuntu@k8s-instance-6:~$ curl http://localhost:8080/info
{"hostname": "7ecc8f714f1b", "ip_address": "172.19.0.2"}
ubuntu@k8s-instance-6:~$

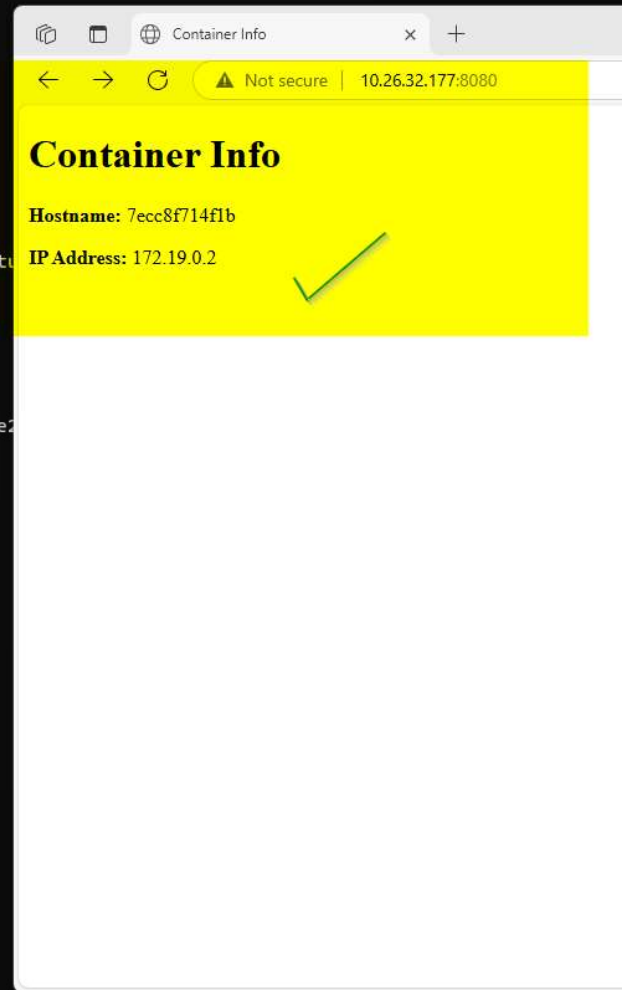
```

```

Step 8/10 : ENV PATH=/root/.local/bin:$PATH
--> Running in 9d500900daba
--> Removed intermediate container 9d500900daba
--> 111fccc9bce6
Step 9/10 : COPY app.py .
--> 4bd4e1ff7390
Step 10/10 : CMD ["python", "app.py"]
--> Running in bc12b60bee20
--> Removed intermediate container bc12b60bee20
--> 55806b7018a8
Successfully built 55806b7018a8
Successfully tagged simple-api:latest
ubuntu@k8s-instance-6:~$ docker run -d \
  --name backend \
  --network mynetwork \
  simple-api
7ecc8f714f1b2453db1d58a2d6036492dee541b561b4508a5b688829f8105543
ubuntu@k8s-instance-6:~$ docker build -t frontend-image ./frontend
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
              Install the buildx component to build images with BuildKit:
              https://docs.docker.com/go/buildx/

Sending build context to Docker daemon  4.608kB
Step 1/5 : FROM nginx:alpine
alpine: Pulling from library/nginx
Digest: sha256:65645c7bb6a0661892a8b03b89d0743208a18dd2f3f17a54ef4b76fb8e2
Status: Downloaded newer image for nginx:alpine
--> 6769dc3a703c
Step 2/5 : WORKDIR /usr/share/nginx/html
--> Running in e33d0295f8e4
--> Removed intermediate container e33d0295f8e4
--> 70534705d371
Step 3/5 : COPY nginx.conf /etc/nginx/nginx.conf
--> 801c84be4147
Step 4/5 : COPY index.html .
--> 984309d4a417
Step 5/5 : CMD ["nginx", "-g", "daemon off;"]
--> Running in 5046cdcdbdb07
--> Removed intermediate container 5046cdcdbdb07
--> 9466c36190a8
Successfully built 9466c36190a8
Successfully tagged frontend-image:latest
ubuntu@k8s-instance-6:~$ docker run -d \
  --name frontend \
  --network mynetwork \
  -p 8080:80 \
  frontend-image
f3b5cb5a03c68fe2fed7c0b6ff7d00dd5e956a400a4b3e7a688a100b5bef613d
ubuntu@k8s-instance-6:~$

```



Exercise 3: The `.dockerignore` File

- Create a `.dockerignore` file in both Back End and Front End repos to exclude **all unnecessary** files and directories when building.

```
ubuntu@k8s-instance-6:~/backend$ ll
total 28
drwxrwxr-x  2 ubuntu ubuntu 4096 May 11 18:50 ./
drwxr-x--- 14 ubuntu ubuntu 4096 May 11 18:58 ../
-rw-rw-r--  1 ubuntu ubuntu  172 May 11 18:50 .dockerignore
-rw-rw-r--  1 ubuntu ubuntu  315 May 11 18:22 Dockerfile
-rw-rw-r--  1 ubuntu ubuntu  424 May 11 18:20 app.py
-rw-rw-r--  1 ubuntu ubuntu   6 May 11 17:43 requirement.txt
-rw-rw-r--  1 ubuntu ubuntu  17 May 11 18:20 requirements.txt
ubuntu@k8s-instance-6:~/backend$ cat .dockerignore
# Sistema y metadatos
.git
.gitignore

# Archivos Python compilados
__pycache__/
*$py.class

# Entornos virtuales
venv/
env/
.env

# Docker
Dockerfile*
docker-compose.yml
ubuntu@k8s-instance-6:~/backend$
```



```

ubuntu@k8s-instance-6:~/frontend$ ll
total 24
drwxrwxr-x  2 ubuntu ubuntu 4096 May 11 18:51 ./
drwxr-x--- 14 ubuntu ubuntu 4096 May 11 18:58 ../
-rw-rw-r--  1 ubuntu ubuntu  191 May 11 18:51 .dockerignore
-rw-rw-r--  1 ubuntu ubuntu  182 May 11 18:19 Dockerfile
-rw-rw-r--  1 ubuntu ubuntu  657 May 11 18:18 index.html
-rw-rw-r--  1 ubuntu ubuntu  297 May 11 18:18 nginx.conf
ubuntu@k8s-instance-6:~/frontend$ cat .dockerignore
# Sistema y metadatos
.git
.gitignore

# Dependencias y builds locales
build/

# Archivos de configuración local
.env
.vscode/
.idea/

# Logs
*.log

# Docker
Dockerfile*
docker-compose.yml
ubuntu@k8s-instance-6:~/frontend$

```

Exercise 4: Private Registry

- Push the previously built Back End and Front End images to the private registry at `docker.jala.pro`.
- Tag your images: `docker.jala.pro/docker-training/[CONTAINER-NAME=BackEnd || FrontEnd]:[TAG=FullName]`
For Instance: `docker.jala.pro/docker-training/backend:calebespinoza`

```
ubuntu@k8s-instance-6:~$ docker login docker.jala.pro
```

```
Username: robot$docker-training+noemi.guzman
```

```
Password:
```

```
WARNING! Your credentials are stored unencrypted in '/home/ubuntu/.docker/config.js'
Configure a credential helper to remove this warning. See
https://docs.docker.com/go/credential-store/
```

```
Login Succeeded
```

```
ubuntu@k8s-instance-6:~$ docker tag simple-api \
    docker.jala.pro/docker-training/backend:calebespinoza
```

```
ubuntu@k8s-instance-6:~$ docker tag frontend-image \
```

```
ubuntu@k8s-instance-6:~$ docker tag simple-api docker.jala.pro/docker-training/backend:calebespinoza
```

```
ubuntu@k8s-instance-6:~$ docker tag frontend-image \
    docker.jala.pro/docker-training/frontend:calebespinoza
```

```
ubuntu@k8s-instance-6:~$
```

```
ubuntu@k8s-instance-6:~$ docker push docker.jala.pro/docker-training/backend:calebespinoza
```

```
The push refers to repository [docker.jala.pro/docker-training/backend]
```

```
2601e750c1aa: Pushed
```

```
aa8bca50c27a: Pushed
```

```
b7fcc45e3071: Pushed
```

```
23aa89a8a424: Pushed
```

```
91bd78b864ed: Pushed
```

```
adb057d02f88: Pushed
```

```
6c4c763d22d0: Pushed
```

```
calebespinoza: digest: sha256:fea5f51e798e179797d3a010c9da456ff6615822739a87547cbbcb488ae24fd5 size: 1783
```

```
ubuntu@k8s-instance-6:~$ docker push docker.jala.pro/docker-training/frontend:calebespinoza
```

```
The push refers to repository [docker.jala.pro/docker-training/frontend]
```

```
8362291027a6: Pushed
```

```
d462931dd2bb: Pushed
```

```
0d853d50b128: Pushed
```

```
947e805a4ac7: Pushed
```

```
811a4dbbf4a5: Pushed
```

```
b8d7d1d22634: Pushed
```

```
e244aa659f61: Pushed
```

```
c56f134d3805: Pushed
```

```
d71eae0084c1: Pushed
```

```
08000c18d16d: Pushed
```

```
calebespinoza: digest: sha256:52275d3537fb2be091f069bc59791316456f01e16facaf269d03748aef2db7b8 size: 2403
```

```
ubuntu@k8s-instance-6:~$
```

```
ubuntu@k8s-instance-6:~$ docker pull docker.jala.pro/docker-training/backend:calebespinoza
```

```
calebespinoza: Pulling from docker-training/backend
```

```
Digest: sha256:fea5f51e798e179797d3a010c9da456ff6615822739a87547cbbcb488ae24fd5
```

```
Status: Image is up to date for docker.jala.pro/docker-training/backend:calebespinoza
```

```
docker.jala.pro/docker-training/backend:calebespinoza
```

```
ubuntu@k8s-instance-6:~$ docker pull docker.jala.pro/docker-training/frontend:calebespinoza
```

```
calebespinoza: Pulling from docker-training/frontend
```

```
Digest: sha256:52275d3537fb2be091f069bc59791316456f01e16facaf269d03748aef2db7b8
```

```
Status: Image is up to date for docker.jala.pro/docker-training/frontend:calebespinoza
```

```
docker.jala.pro/docker-training/frontend:calebespinoza
```

```
ubuntu@k8s-instance-6:~$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
docker.jala.pro/docker-training/frontend	calebespinoza	9466c36190a8	44 minutes ago	48.2MB
frontend-image	latest	9466c36190a8	44 minutes ago	48.2MB
simple-api	latest	55806b7018a8	45 minutes ago	135MB
docker.jala.pro/docker-training/backend	calebespinoza	55806b7018a8	45 minutes ago	135MB
<none>	<none>	fd687ad173ac	45 minutes ago	146MB
<none>	<none>	ff0f65f02ff4	About an hour ago	48.2MB
<none>	<none>	d13cca37277c	About an hour ago	135MB
python	3.11-slim	bea8499a31bb	3 days ago	130MB
nginx	alpine	6769dc3a703c	3 weeks ago	48.2MB
mariadb	latest	9f3d79eba61e	2 months ago	328MB
phpmyadmin	latest	7a9597477072	3 months ago	570MB

```
ubuntu@k8s-instance-6:~$
```

```
ubuntu@k8s-instance-6:~$ docker pull docker.jala.pro/docker-training/backend:calebespinoza
calebespinoza: Pulling from docker-training/backend
Digest: sha256:fea5f51e798e179797d3a010c9da456ff6615822739a87547cbbcb488ae24fd5
Status: Image is up to date for docker.jala.pro/docker-training/backend:calebespinoza
docker.jala.pro/docker-training/backend:calebespinoza
```

```
ubuntu@k8s-instance-6:~$ docker pull docker.jala.pro/docker-training/frontend:calebespinoza
calebespinoza: Pulling from docker-training/frontend
Digest: sha256:52275d3537fb2be091f069bc59791316456f01e16facaf269d03748aef2db7b8
Status: Image is up to date for docker.jala.pro/docker-training/frontend:calebespinoza
docker.jala.pro/docker-training/frontend:calebespinoza
```

```
ubuntu@k8s-instance-6:~$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
docker.jala.pro/docker-training/frontend	calebespinoza	9466c36190a8	44 minutes ago	48.2MB
frontend-image	latest	9466c36190a8	44 minutes ago	48.2MB
simple-api	latest	55806b7018a8	45 minutes ago	135MB
docker.jala.pro/docker-training/backend	calebespinoza	55806b7018a8	45 minutes ago	135MB
<none>	<none>	fd687ad173ac	45 minutes ago	146MB
<none>	<none>	ff0f65f02ff4	About an hour ago	48.2MB
<none>	<none>	d13cca37277c	About an hour ago	135MB
python	3.11-slim	bea8499a31bb	3 days ago	130MB
nginx	alpine	6769dc3a703c	3 weeks ago	48.2MB
mariadb	latest	9f3d79eba61e	2 months ago	328MB
phpmyadmin	latest	7a9597477072	3 months ago	570MB

```
ubuntu@k8s-instance-6:~$
```