

Rafael Andres Alvarez Mazariegos 1018419

Teerraform Cloud

Paso 1

App.py

A screenshot of a code editor window with a dark theme. The window has multiple tabs at the top, with the active tab labeled 'app.py'. The editor displays Python code for a Flask application. The code imports Flask, datetime, and pytz, creates a Flask app instance, defines a route for the root path that returns the current datetime in a specific timezone, and includes a main block to run the app on all interfaces at port 5000.

```
from flask import Flask
from datetime import datetime
import pytz

app = Flask(__name__)

@app.route('/')
def get_datetime():
    tz = pytz.timezone('UTC') # Puedes ajustar a tu zona horaria preferida
    now = datetime.now(tz)
    return {"datetime": now.isoformat(), "timezone": str(now.tzinfo)}

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000)
```

Docker File

```
File Edit View
# Dockerfile
FROM python:3.9-slim
RUN pip install Flask pytz
COPY app.py /app.py
ENTRYPOINT ["python"]
CMD ["app.py"]
```

Docker

docker desktop






Search for images, containers, volumes, ... **Ctrl+K**

Containers [Give feedback](#)

Container CPU usage ⓘ Container memory usage ⓘ [Show charts](#)

No containers are running.

Search Only show running containers ☐

<input type="checkbox"/>	Name	Image	Status	CPU (%)	Port(s)	Last started	Actions
<input type="checkbox"/>	 elated_swan: c68dd32f35a7	datetime-api	Exited (1)	N/A	5000:5000 	1 minute ago	  

Showing 1 item

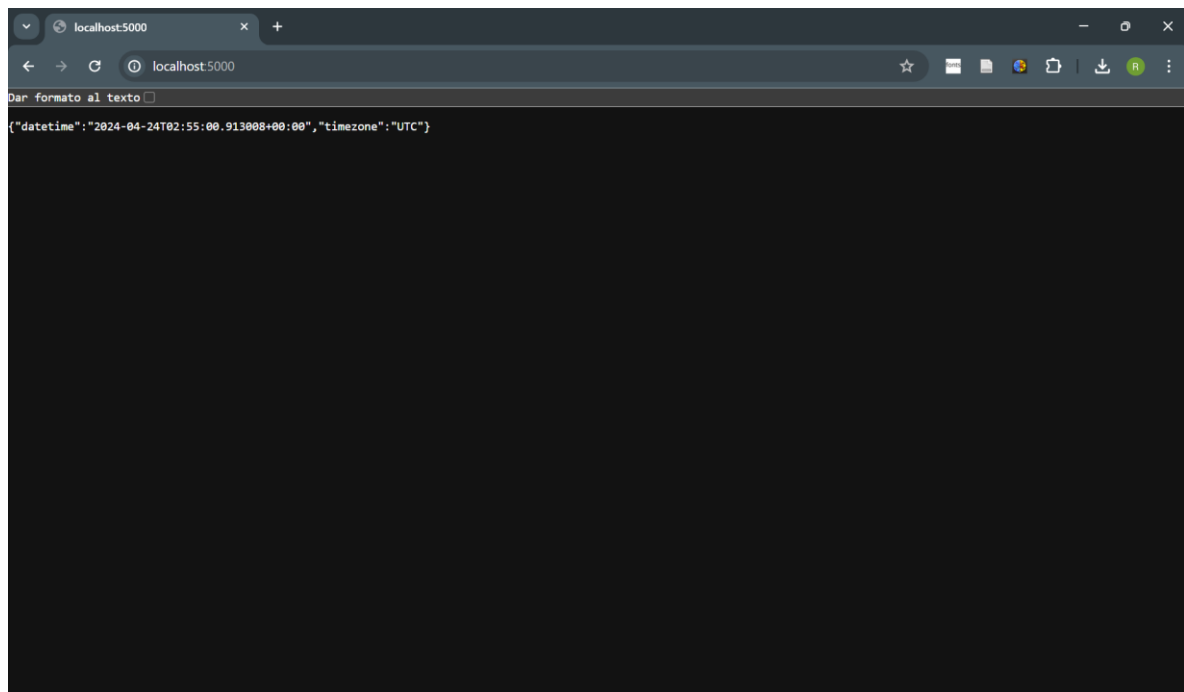
```
ca) Administrador: Símbolo del sistema - docker run -p 5000:5000 datetime-api
=> [internal] load build definition from Dockerfile                                0.0s
=> => transferring dockerfile: 159B                                              0.0s
=> [internal] load metadata for docker.io/library/python:3.9-slim                0.7s
=> [auth] library/python:pull token for registry-1.docker.io                    0.0s
=> [internal] load .dockerignore                                                  0.0s
=> => transferring context: 2B                                                    0.0s
=> [1/3] FROM docker.io/library/python:3.9-slim@sha256:fd8693fc6115345452a19654c1388bc9a1029cda7b98e1bce45ede67e 0.0s
=> [internal] load build context                                                  0.0s
=> => transferring context: 415B                                                  0.0s
=> CACHED [2/3] RUN pip install Flask pytz                                       0.0s
=> [3/3] COPY app.py /app.py                                                     0.0s
=> exporting to image                                                            0.0s
=> => exporting layers                                                            0.0s
=> => writing image sha256:45487dfce64414bd95e6a5ed9ab2b1ec9b3eaf67c831785e5550286e44e9e2b0 0.0s
=> => naming to docker.io/library/datetime-api                                  0.0s

View build details: docker-desktop://dashboard/build/default/default/i9nuqw9iwre8o0vivbo2yx3li

What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview

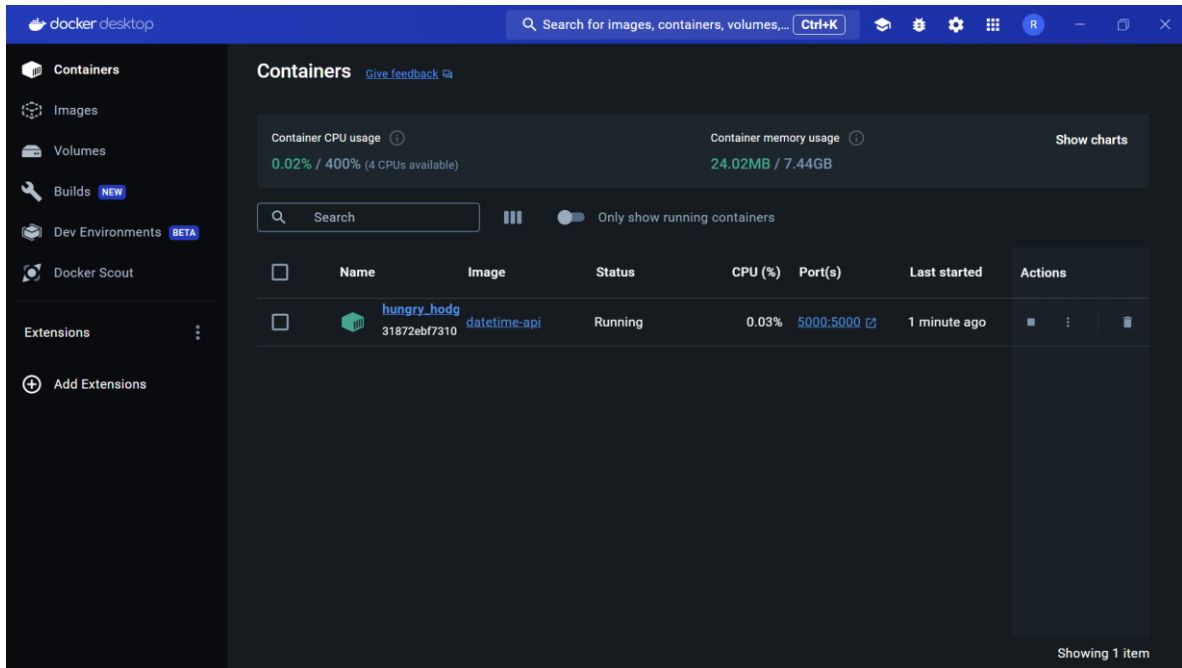
C:\Users\Usuario Dell\Desktop\terraform_1.8.0_windows_amd64>docker run -p 5000:5000 datetime-api
* Serving Flask app 'app'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://172.17.0.2:5000
Press CTRL+C to quit
```

Localhost



```
localhost:5000
{"datetime": "2024-04-24T02:55:00.913008+00:00", "timezone": "UTC"}
```

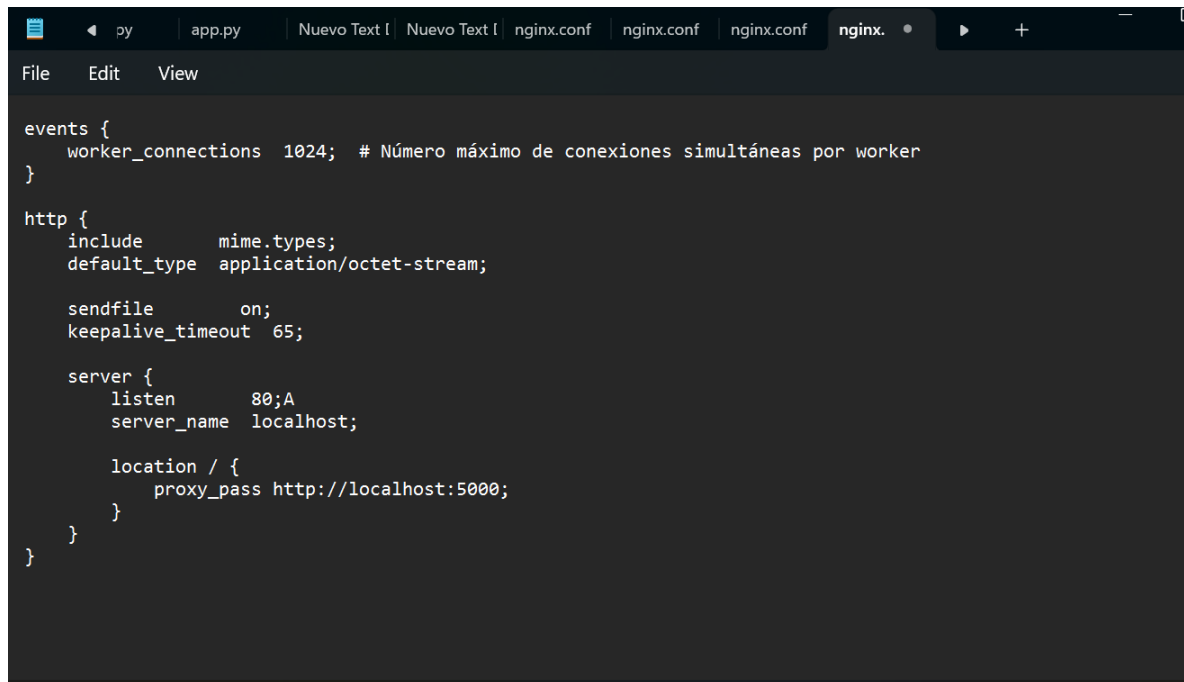
Docker corriendo



Nginx

```
Administrator: Símbolo del sistema - nginx -c nginx.conf
Microsoft Windows [Versión 10.0.22631.3447]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Windows\System32>cd C:\Users\Usuario Dell\Desktop\terraform_1.8.0_windows_amd64\nnginx-1.26.0\nnginx-1.26.0
C:\Users\Usuario Dell\Desktop\terraform_1.8.0_windows_amd64\nnginx-1.26.0\nnginx-1.26.0>nginx -c nginx.conf
```



The image shows a code editor window with a dark theme. The top bar contains several tabs: 'py', 'app.py', 'Nuevo Text I', 'Nuevo Text I', 'nginx.conf', 'nginx.conf', 'nginx.conf', and 'nginx.'. The 'nginx.' tab is currently active. Below the tabs is a menu bar with 'File', 'Edit', and 'View'. The main area of the editor displays the configuration for an nginx.conf file. The code is as follows:

```
events {  
    worker_connections 1024; # Número máximo de conexiones simultáneas por worker  
}  
  
http {  
    include mime.types;  
    default_type application/octet-stream;  
  
    sendfile on;  
    keepalive_timeout 65;  
  
    server {  
        listen 80;A  
        server_name localhost;  
  
        location / {  
            proxy_pass http://localhost:5000;  
        }  
    }  
}
```