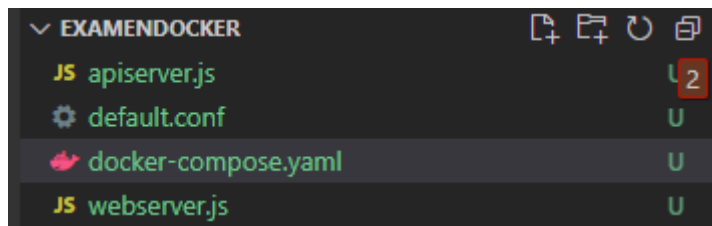


Tenemos una estructura como la siguiente:

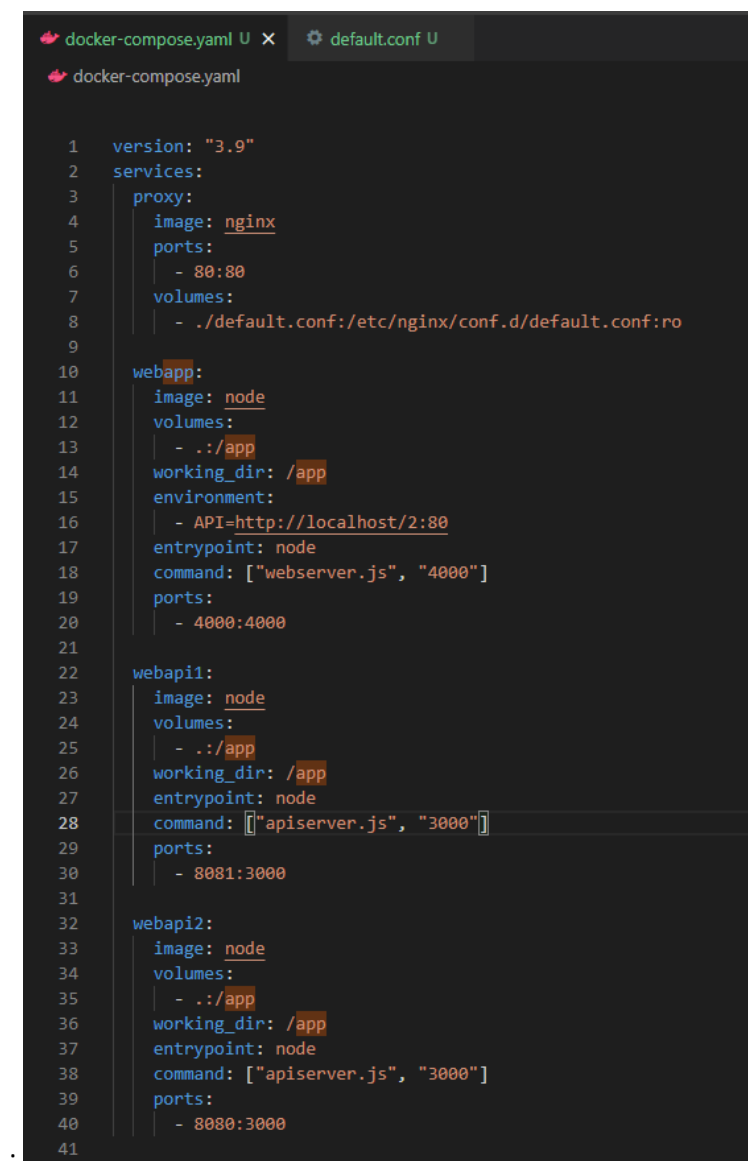


Apiserver.js y webserver.js son los descargados del enunciado.

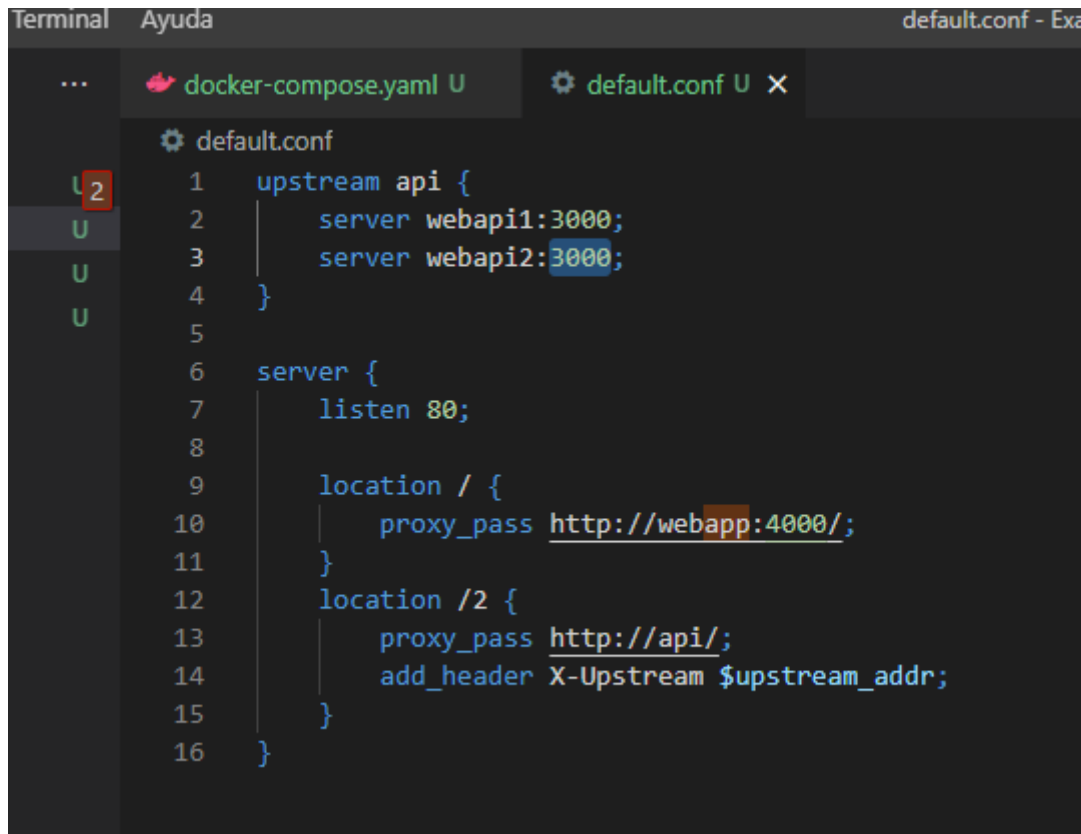
Examen

El docker compose es como el siguiente

Docker-compose.yaml



Creamos un volumen en el que guardamos la configuración del proxy



```
1 upstream api {
2     server webapi1:3000;
3     server webapi2:3000;
4 }
5
6 server {
7     listen 80;
8
9     location / {
10        proxy_pass http://webapp:4000/;
11    }
12    location /2 {
13        proxy_pass http://api/;
14        add_header X-Upstream $upstream_addr;
15    }
16 }
```

Ejecutamos el docker compose, con `docker compose up`, y estará nuestra web en la dirección

<http://localhost:4000/>

Aparece lo siguiente



Llamamos a la api y funciona, además cambiando entre las dos apis, viendo también que el balanceador de carga funciona correctamente

Cursor: 21_22_INFOR_2DAW3_DS: x

21_22_INFOR_2DAW3_DSP: EVAL: x

localhost:4000

x

+

<

>

↺

localhost:4000

web: f828af59d300 12:21:24 464

Call API

- {"api":"2ecc608ffe2e:3000","time":"12:21:36 360","ipreq":"172.26.0.4"}
- {"api":"a5181df689a3:3000","time":"12:21:36 558","ipreq":"172.26.0.4"}
- {"api":"2ecc608ffe2e:3000","time":"12:21:36 742","ipreq":"172.26.0.4"}
- {"api":"a5181df689a3:3000","time":"12:21:36 902","ipreq":"172.26.0.4"}
- {"api":"2ecc608ffe2e:3000","time":"12:21:37 102","ipreq":"172.26.0.4"}
- {"api":"a5181df689a3:3000","time":"12:21:37 318","ipreq":"172.26.0.4"}