Deployment of a Docker container of "Yolo" in two cloud services: as an Azure container instance and a virtual machine

José Luis Puente Bodoque

June 25, 2023

Abstract

In this report I comment the steps followed in order to deploy and test a Yolo app project as mandatory assignment in module 13: Cloud Computing.

Contents

1 Testing the app locally

To test the app locally, I open a terminal window and enter the following command to run the app.

```
python app.py
```

Once the app is running I can test its functionality. I can access it at:

```
localhost:4000\upload
```

For some Real Madrid starter line-up image in JPEG format, the result is the one below.

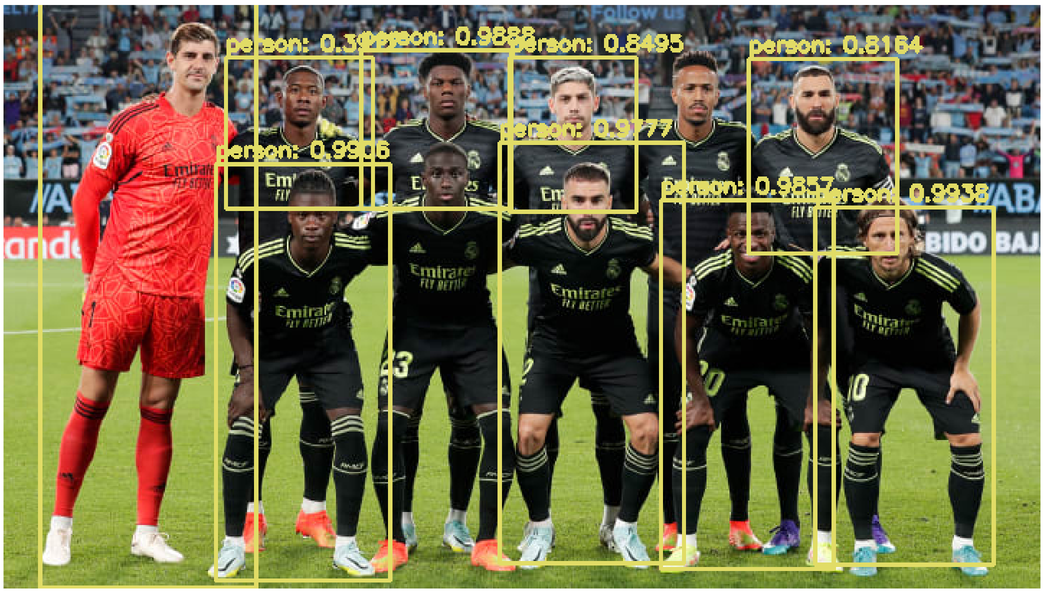


Figure 1: Testing the app locally

2 Building the container

The project already provides a Dockerfile in the same directory as the app.py so I just run these Docker commands:

```
# Set the default Docker context
docker context use default
```

```
# Build the container
docker build -t yolo .
```

I received the error "failed to fetch from deb.debian.org…" when I entered the second command. I solved it[1] by entering the following command to clear the Docker cache:

```
docker system prune --all
```

3 Testing the container locally

To verify that the container is working I entered the command:

```
# Verify that the container is working
docker run -p 4000:4000 yolo
```

Then I tested it at:

```
http://127.0.0.1:4000/upload
```

4 Deployment in Azure as an Azure container instance

I have deployed a container from one already pushed to hub.docker.com. So the first step is to create an account in hub.docker.com and then create a repository.

4.1 Pushing my container to hub.docker.com

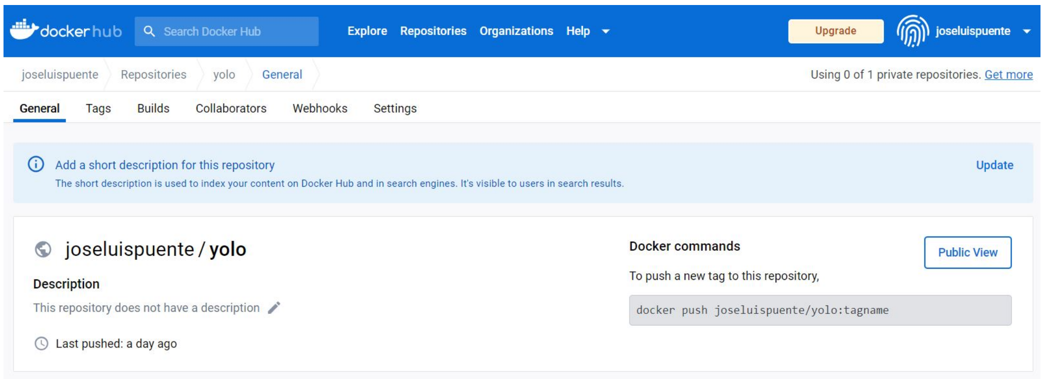
My public repository for yolo project is the one below.

In order to push my container to my repository and then test it locally, I opened a window terminal in the same directory as the Yolo app project and entered:

```
# Tag my image:
docker tag yolo joseluispuente/yolo:some-version

# Upload it to hub.docker.com:
```

[1]This answer truly deserves all the merit.


docker hub | Search Docker Hub | Explore | Repositories | Organizations | Help | Upgrade | joseluispuente

joseluispuente Repositories yolo General

Using 0 of 1 private repositories. Get more

General | Tags | Builds | Collaborators | Webhooks | Settings

Add a short description for this repository

The short description is used to index your content on Docker Hub and in search engines. It's visible to users in search results.

Update

joseluispuente / yolo

Description

This repository does not have a description

Last pushed: a day ago

Docker commands

Public View

To push a new tag to this repository,

```
docker push joseluispuente/yolo:tagname
```

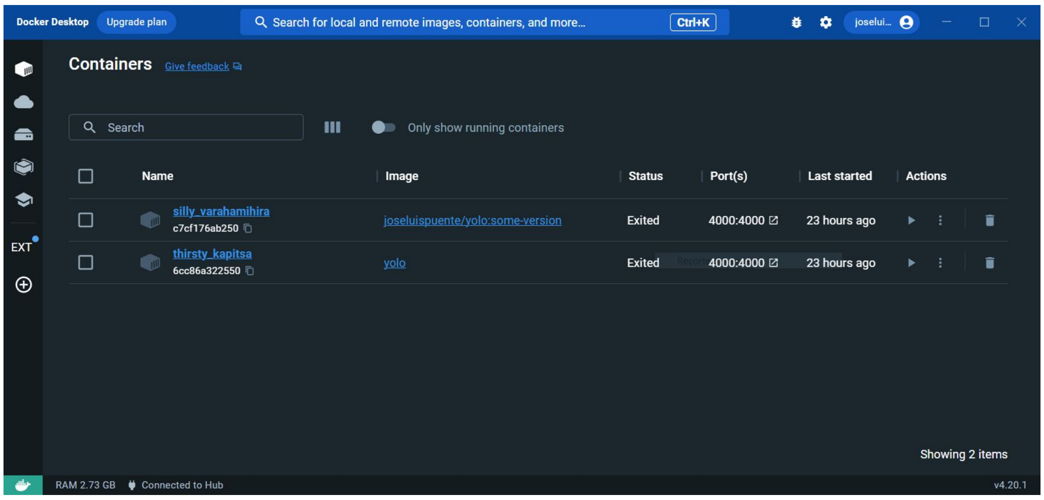
Figure 2: My repository at hub.docker.com

```
docker push joseluispuente/yolo:some-version

# Run it just by typing (nameless container):
docker run -p 4000:4000 joseluispuente/yolo:some-version

# Test it at:
localhost:4000/upload
```

I can see that my containers are built and visible on Docker Desktop.


Docker Desktop | Upgrade plan | Search for local and remote images, containers, and more... | Ctrl+K | joselui...

Containers | Give feedback

Search | Only show running containers

Name	Image	Status	Port(s)	Last started	Actions
silly_varahamihira c7cf176ab250	joseluispuente/yolo:some-version	Exited	4000:4000	23 hours ago	
thirsty_kapitsa 6cc86a322550	yolo	Exited	4000:4000	23 hours ago	

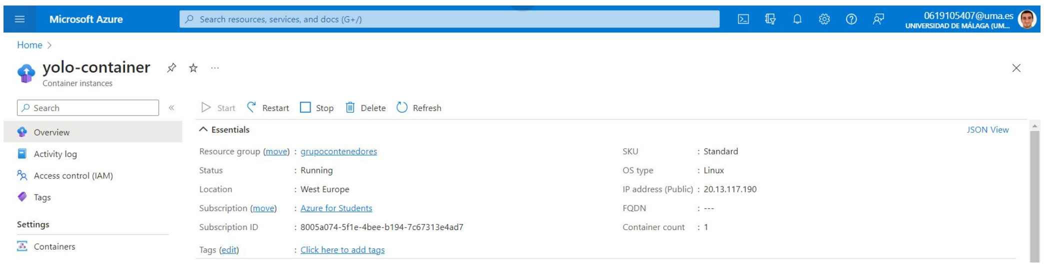
Showing 2 items

RAM 2.73 GB | Connected to Hub | v4.20.1

Figure 3: My containers on Docker Desktop

4.2 Deployment and test of an Azure Container Instance

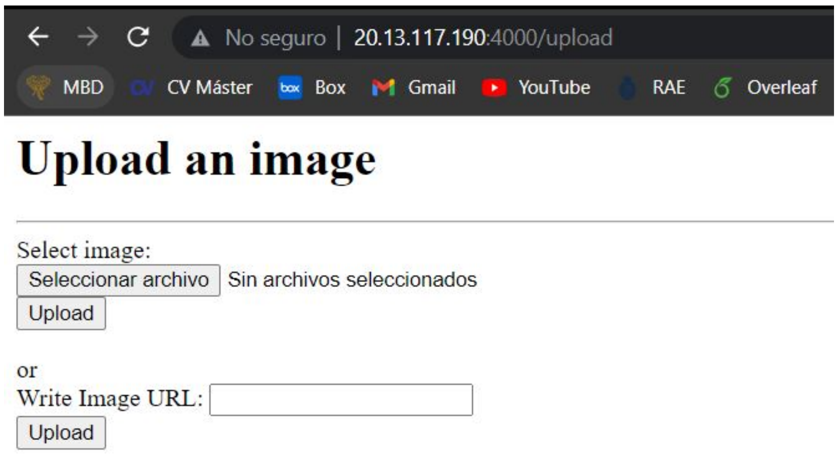
I have created a new Azure container instance by following the steps indicated in class slides. Then I click on Run and the resulting window is shown below. Note the public IP is 20.13.117.190.



Resource group (move)	: grupocontenedores	SKU	: Standard
Status	: Running	OS type	: Linux
Location	: West Europe	IP address (Public)	: 20.13.117.190
Subscription (move)	: Azure for Students	FQDN	: ---
Subscription ID	: 8005a074-5f1e-4bee-b194-7c67313e4ad7	Container count	: 1
Tags (edit)	: Click here to add tags		

Figure 4: My Azure container instance of Yolo

Finally I may test the running container by using its public IP and port number 4000.



Upload an image

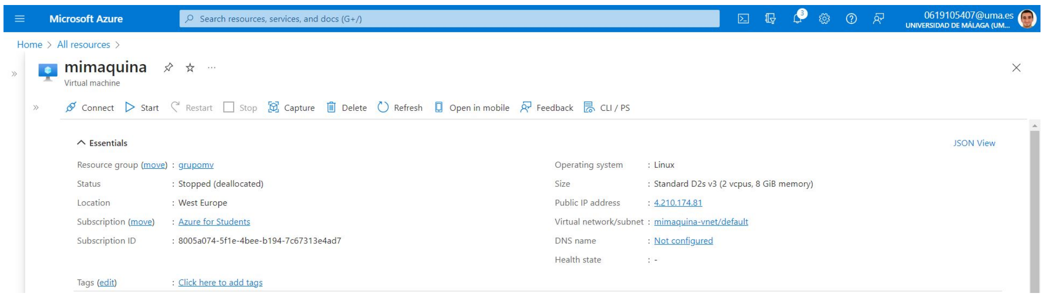
Select image:
Seleccionar archivo Sin archivos seleccionados
Upload

or
Write Image URL:
Upload

Figure 5: Test of y Azure container instance of Yolo

4.3 Deployment and test in a virtual machine

First step is to create a new virtual machine as resource in Azure.



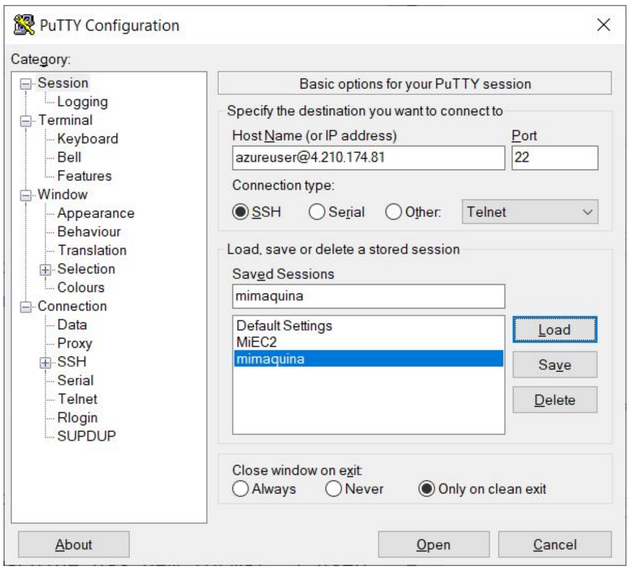
mimaquina
Virtual machine

Essentials

Resource group (move)	: grupomv	Operating system	: Linux
Status	: Stopped (deallocated)	Size	: Standard D2s v3 (2 vcpus, 8 GiB memory)
Location	: West Europe	Public IP address	: 4.210.174.81
Subscription (move)	: Azure for Students	Virtual network/subnet	: mimaquina-vnet/default
Subscription ID	: 8005a074-5f1e-4bee-b194-7c67313e4ad7	DNS name	: Not configured
		Health state	: -
Tags (edit)	: Click here to add tags		

Figure 6: My virtual machine

Next step is to connect to it using PuTTY with the user `userazure` and public IP 4.210.174.81, so that the host name to type in the logging tap is `userazure@4.210.174.81`. The private key must suit PuTTY format, that is, ppk (PuTTY Private Key) format. As the key I downloaded when I created my virtual machine has pem format, I used PuTTygen to convert it to ppk format.



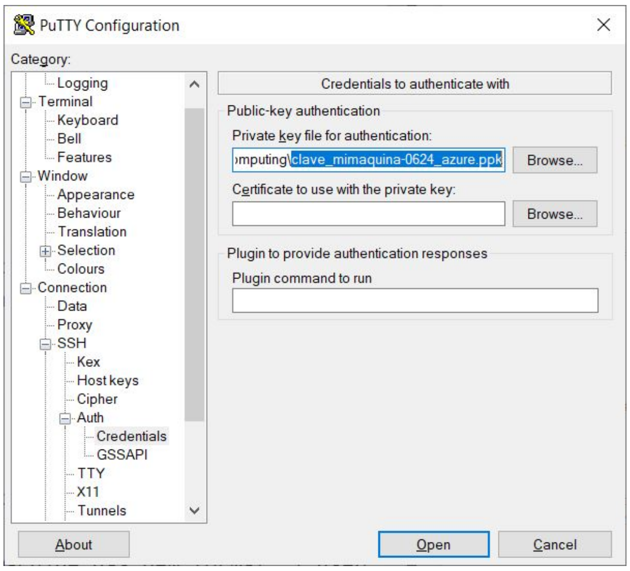
Specify the destination you want to connect to
Host Name (or IP address): azureuser@4.210.174.81
Port: 22
Connection type: [✓ SSH] Serial Other: Telnet
Load, save or delete a stored session
Saved Sessions: mimaquina
Default Settings
MiEC2
mimaquina
Close window on exit: Always Never [✓ Only on clean exit]

Figure 7: Putty login

Once I was already connected I installed Python at first.

```
sudo apt-get install python
```

Then I tried to create a virtual environment, but a helpful error message suggested me to install `python3-venv` before.



Credentials to authenticate with

Public-key authentication

Private key file for authentication: ›mputing\clave_mimaquina-0624_azure.ppk

Certificate to use with the private key:

Plugin to provide authentication responses

Plugin command to run

Figure 8: My PuTTY private key imported

```
sudo apt-get install python3-venv
```

Now I could create a virtual environment (.venv):

```
python3 -m venv .venv
```

And I activated it:

```
source .venv/bin/activate
```

Now I installed all dependencies specified in `requirements.txt`:

```
python -m pip install -r requirements.txt
```

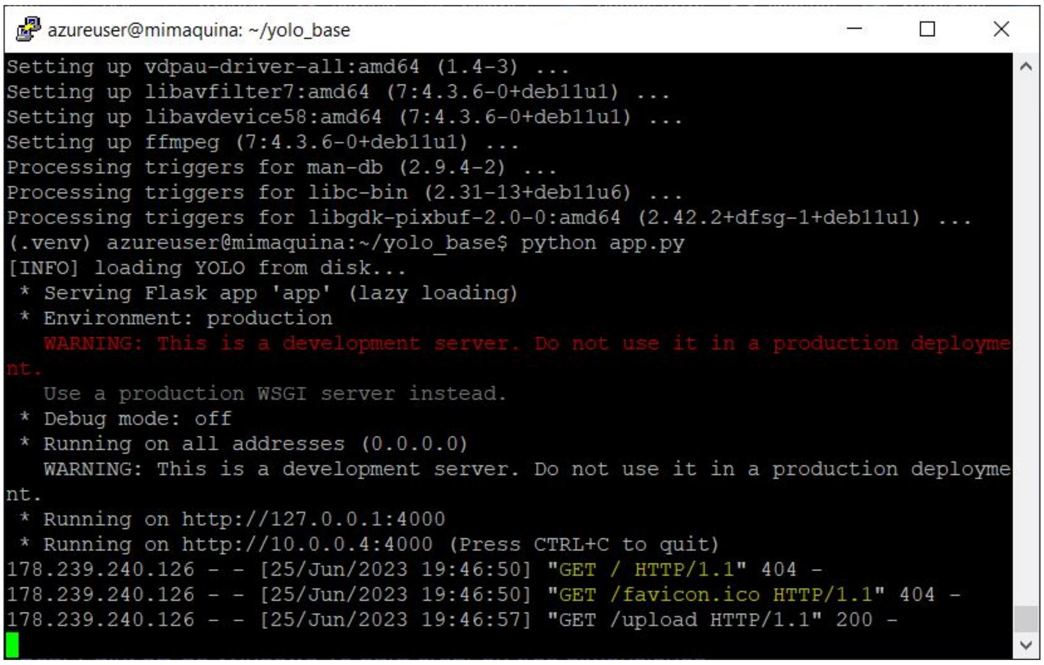
And when I ran the app.py I received an error related to the `cv2` package. I found the solution in this way:

```
sudo apt-get update
sudo apt-get install ffmpeg libsm6 libxext6 -y
```

These two last commands install the `cv2` dependencies that aren't present just by installing the `opencv-python` library.

Now I could successfully launch the app.py.

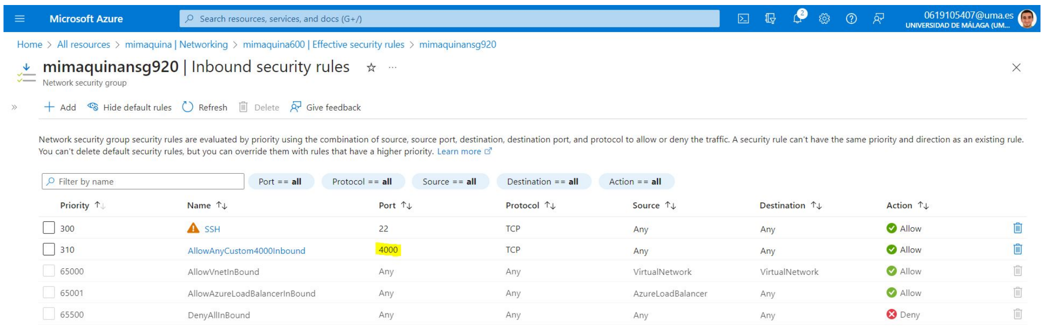
In order to enter from the address bar of the browser it was necessary for me to open the port 4000 in the virtual machine. To do that, I went to



```
Setting up vdpau-driver-all:amd64 (1.4-3) ...
Setting up libavfilter7:amd64 (7:4.3.6-0+deb11u1) ...
Setting up libavdevice58:amd64 (7:4.3.6-0+deb11u1) ...
Setting up ffmpeg (7:4.3.6-0+deb11u1) ...
Processing triggers for man-db (2.9.4-2) ...
Processing triggers for libc-bin (2.31-13+deb11u6) ...
Processing triggers for libgdk-pixbuf-2.0-0:amd64 (2.42.2+dfsg-1+deb11u1) ...
(.venv) azureuser@mimaquina:~/yolo_base$ python app.py
[INFO] loading YOLO from disk...
 * Serving Flask app 'app' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployme
nt.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on all addresses (0.0.0.0)
   WARNING: This is a development server. Do not use it in a production deployme
nt.
 * Running on http://127.0.0.1:4000
 * Running on http://10.0.0.4:4000 (Press CTRL+C to quit)
178.239.240.126 - - [25/Jun/2023 19:46:50] "GET / HTTP/1.1" 404 -
178.239.240.126 - - [25/Jun/2023 19:46:50] "GET /favicon.ico HTTP/1.1" 404 -
178.239.240.126 - - [25/Jun/2023 19:46:57] "GET /upload HTTP/1.1" 200 -
```

Figure 9: Running app.py

my virtual machine overview, then Networking and add a new inbound port rule.


Microsoft Azure
Search resources, services, and docs (G+/)
0619105407@uma.es
UNIVERSIDAD DE MÁLAGA (UM...
Home > All resources > mimaquina | Networking > mimaquina600 | Effective security rules > mimaquinansg920
Add
Hide default rules
Refresh
Delete
Give feedback
Filter by name
Port == all
Protocol == all
Source == all
Destination == all
Action == all

mimaquinansg920 | Inbound security rules

Network security group

Network security group security rules are evaluated by priority using the combination of source, source port, destination, destination port, and protocol to allow or deny the traffic. A security rule can't have the same priority and direction as an existing rule. You can't delete default security rules, but you can override them with rules that have a higher priority. Learn more

Priority	Name	Port	Protocol	Source	Destination	Action
300	SSH	22	TCP	Any	Any	Allow
310	AllowAnyCustom4000Inbound	4000	TCP	Any	Any	Allow
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowAzureLoadBalancerInBound	Any	Any	AzureLoadBalancer	Any	Allow
65500	DenyAllInBound	Any	Any	Any	Any	Deny

Figure 10: Inbound port rule

This allowed me to enter the application.

Finally I test it with the same image. Note that the public IP is the virtual machine, not the container instance one.

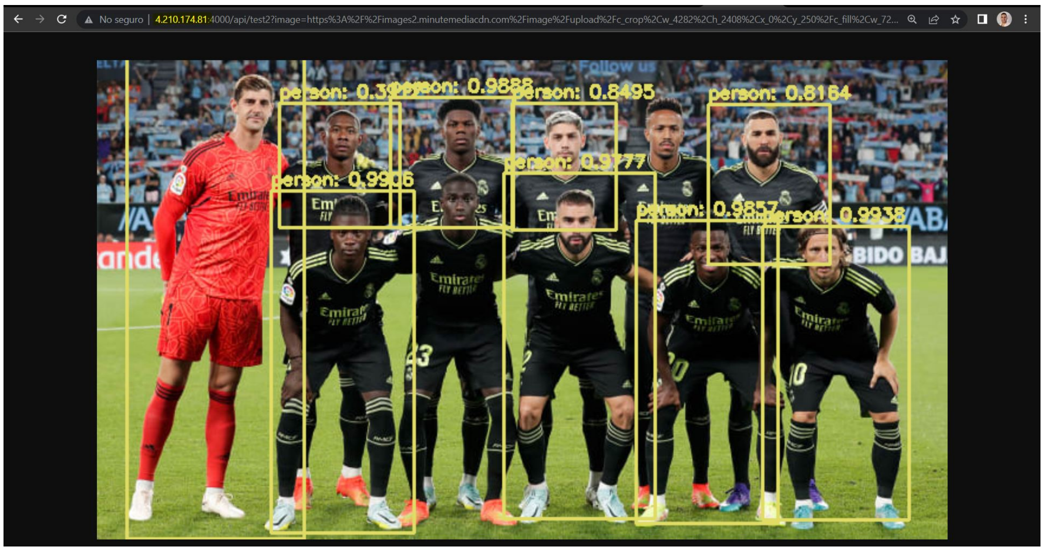

No seguro
4.210.174.81:4000/api/test2?image=https%3A%2F%2Fimages2.minutemediacdn.com%2Fimage%2Fupload%2Fc_crop%2Cw_4282%2Ch_2408%2Cx_0%2Cy_250%2Fc_fill%2Cw_72...
person: 0.3
person: 0.9888
person: 0.8495
person: 0.8164
person: 0.9906
person: 0.9777
person: 0.9857
person: 0.9938

Figure 11: Access to app running on a virtual machine from a browser