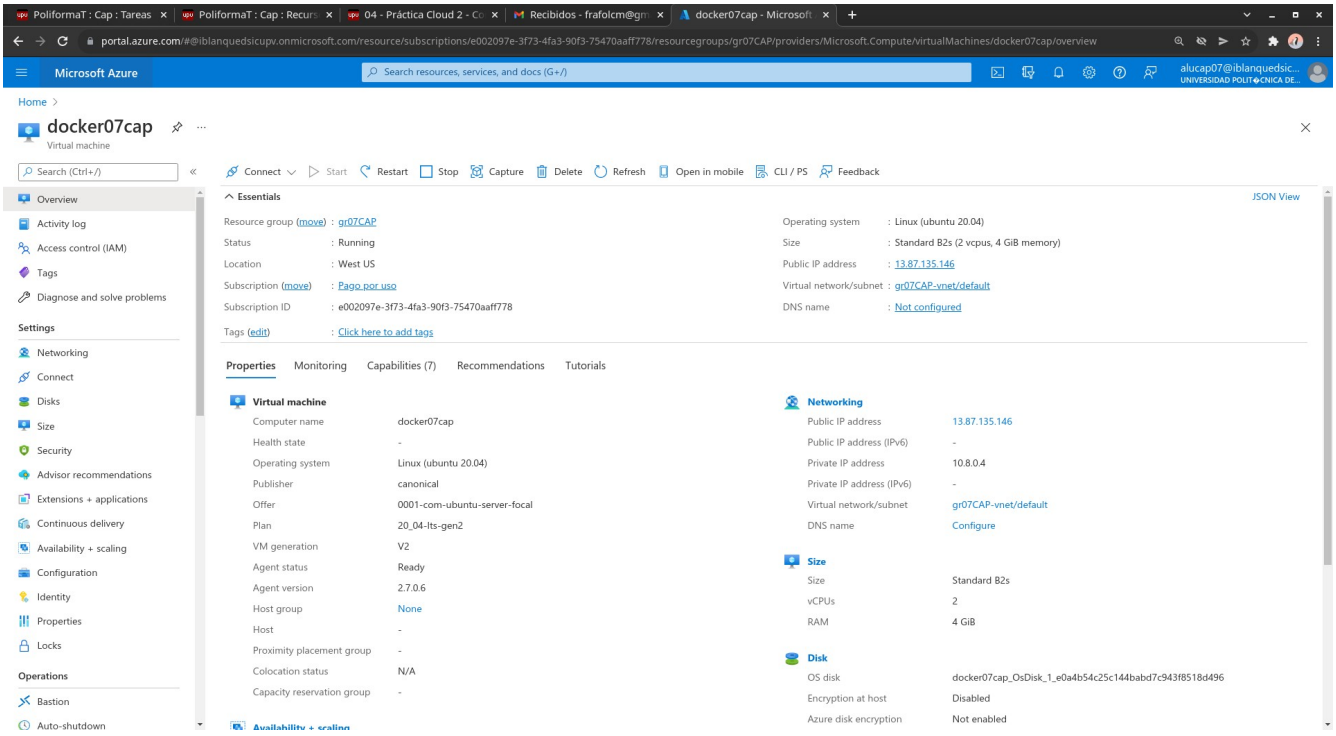
EXAMEN CAP – CLOUD COMPUTING

Francesc Folch Company

a) Crear una máquina virtual Azure e instalar docker

1. Abrimos Azure y seleccionamos *Create Virtual Machine.*
   - resource group: gr07CAP (utilizado en seminarios anteriores).
   - nombre: docker07cap
   - OS: Ubuntu 20.04 LTS
   - Size: standard_b2s
   - User: fran
   - Credentials: (Utilizamos contraseña)
   - disk: standard ssd
   - Seleccionamos siguiente hasta crear la máquina virtual

2. Abrimos los detalles de la máquina virtual



Copiamos la ip y nos conectamos por ssh:

```
$ ssh fran@13.87.135.146
```

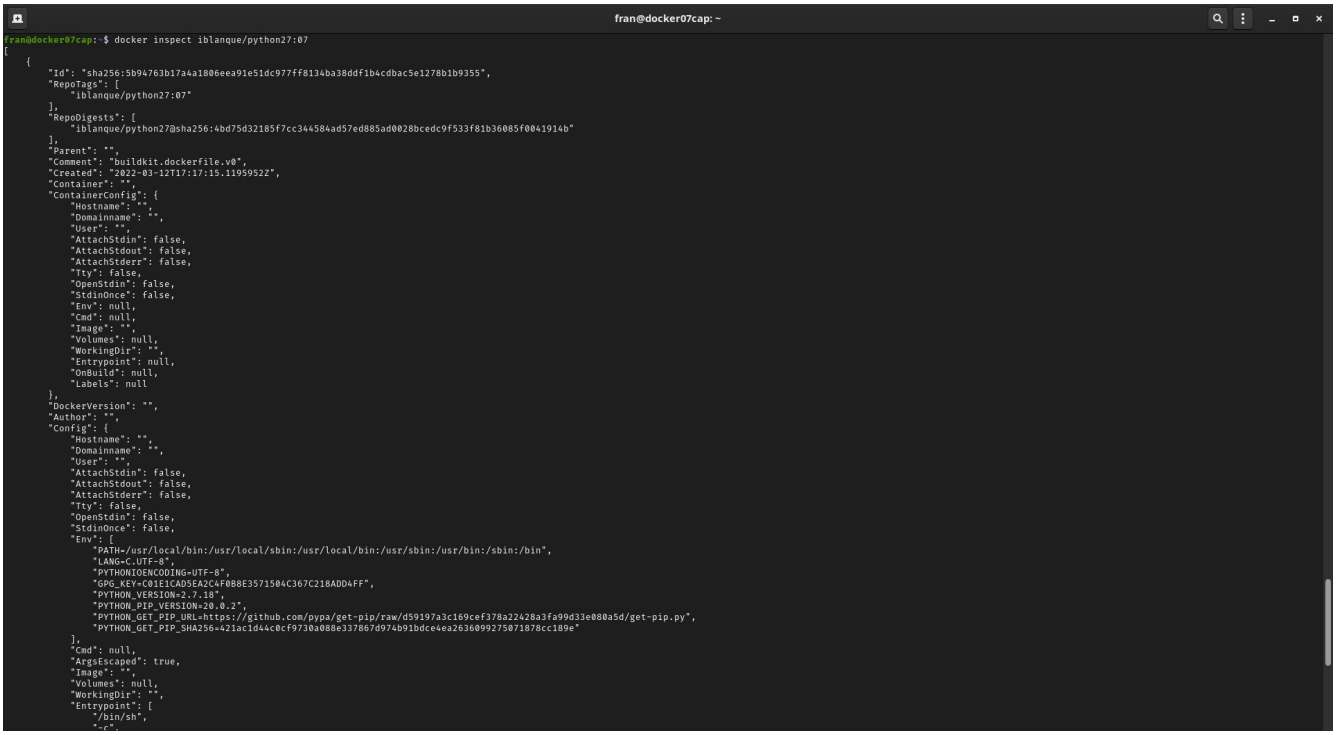
Ejecutamos dentro de la MV:

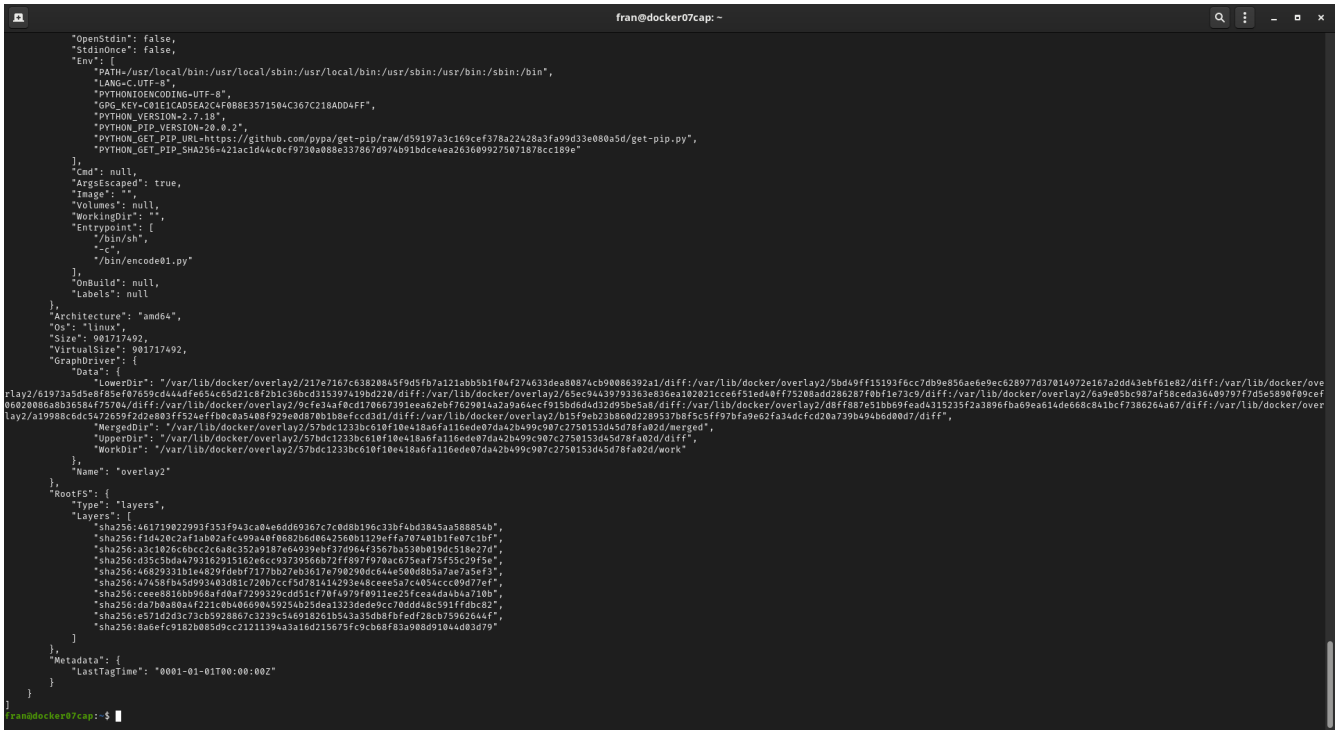
```
$ sudo apt update; sudo apt install apt-transport-https ca-certificates
curl software-properties-common -y

$ curl  -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key
add -
```

```
$ sudo add-apt-repository “deb [arch=amd64]
https://download.docker.com/linux/ubuntu bionic stable”

$ sudo apt update

$ apt-cache policy docker-ce

$ sudo apt install -y docker-ce

$ sudo groupadd docker

$ sudo usermod -aG docker $USER

$ sudo gpasswd -a ${USER} docker

$ sudo service docker restart
```

Hacemos logout y reabrimos la sesión y tendremos docker instalado. Después nos bajamos el contenedor iblanque/python27:07 y lo inspeccionamos

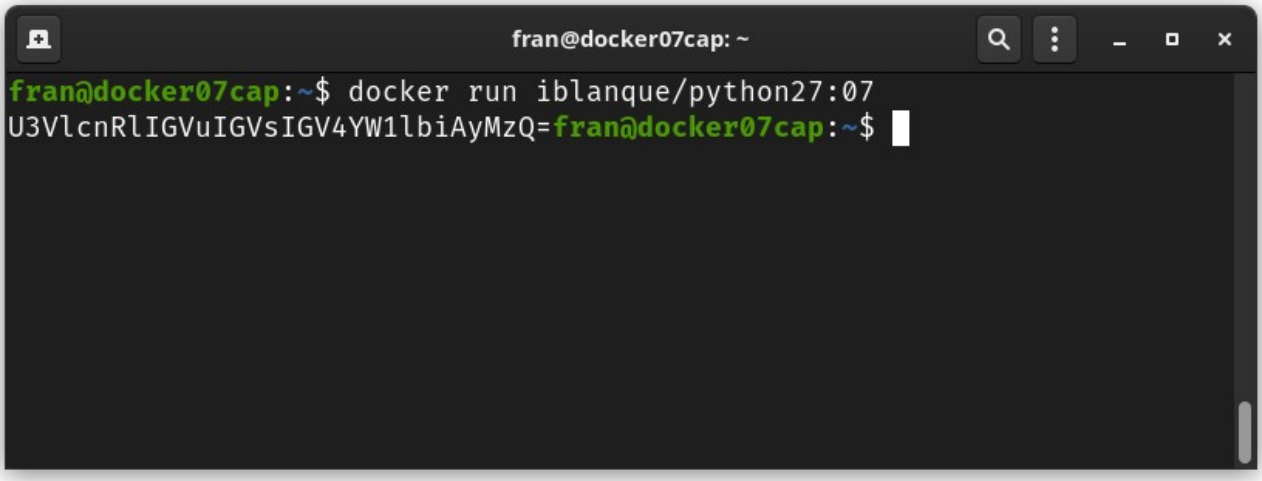
```
$ docker pull iblanque/python27:07
```





Estas capturas son el resultado del comando docker inspect iblanque/python27:07.

b) Ejecutar el contenedor.



Tras terminar se elimina la MV.

c) Crear una función Azure que reciba un parámetro "mensaje" y muestre el mensaje decodificado.

1. En Azure, seleccionamos *Create Function App.*
2. Parámetros:
- Resource group: gr07CAP
- Function APP name: dec07cap
- Publish: seleccionamos *Docker Container*
- Storage Account: gr07capb46c
- Le damos a crear.

3. En mi máquina local (con docker instalado). Ejecuto lo siguiente:

```
$ mkdir $HOME/funciones

$ docker run -it --rm --name azuretools-cli -v $PWD/funciones:/funciones -
p 7071:7071 iblanque/azuretools:20.04 bash

$ cd /funciones

$ func init
     Select a number for worker runtime:
     1. dotnet
     2. dotnet (isolated process)
     3. node
     4. python
     5. powershell
     6. custom
     Choose option: 4
     ...
$ func new
...
9. HTTP Trigger
…
Function name: [HttpTrigger] DecodeFunc
...

$ apt update; apt install vim -y
```

4. Editamos el archivo DecodeFunc/__init__.py
Que contendrá lo siguiente:

```python
import logging, base64

import azure.functions as func


def main(req: func.HttpRequest) -> func.HttpResponse:
    logging.info('Python HTTP trigger function processed a request.')

    name = req.params.get('mensaje')
    if not name:
        try:
            req_body = req.get_json()
        except ValueError:
            pass
        else:
            name = req_body.get('mensaje')

    if name:
        mensaje_decod = base64.b64decode(name)
        return func.HttpResponse(f"{mensaje_decod}")
    else:
        mensaje_decod = base64.b64decode('SG9sYSBtdW5kbw==')
        return func.HttpResponse(f"{mensaje_decod}")
```

Si no le pasamos el parametro mensaje, decodificará un mensaje por defecto, el cual es 'Hola Mundo'.

Después, para testear que funciona ejecutamos el servidor en local:

```
$ func start # terminal 1
```

Desde otra terminal dentro del contenedor, llamamos a la función.

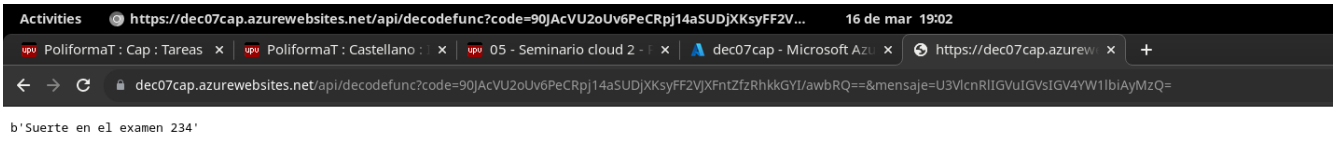
```
$ curl "http://localhost:7071/api/DecodeFunc"
b'Hola mundo'

$ curl "http://localhost:7071/api/DecodeFunc?name=SG9sYSBtdW5kbw=="
b'Hola mundo'
```

Después autenticamos azure y subimos la funcion:

```
$ az login
```

```
$ func azure functionapp publish dec07cap
```

Nos devuelve una url:
https://dec07cap.azurewebsites.net/api/decodefunc?code=90JAcVU2oUv6PeCRpj14aSUDjXKsyFF2VJXFntZfzRhkkGYI/awbRQ==



b'Suerte en el examen 234'