

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

The screenshot shows the Azure portal interface for a virtual machine named 'docker07cap'. The left sidebar contains navigation options like Overview, Activity log, Access control, Tags, and Settings. The main content area is divided into 'Essentials' and 'Properties' sections. The 'Essentials' section shows the resource group 'gr07CAP', status 'Running', location 'West US', subscription 'Pago por uso', and subscription ID 'e002097e-3f73-4fa3-90f3-75470aaf778'. The 'Properties' section is further divided into 'Virtual machine' and 'Networking' tabs. The 'Virtual machine' tab shows details like computer name 'docker07cap', health state, operating system 'Linux (ubuntu 20.04)', publisher 'canonical', offer '0001-com-ubuntu-server-focal', plan '20_04-lts-gen2', VM generation 'V2', agent status 'Ready', agent version '2.7.0.6', host group 'None', host, proximity placement group, colocation status 'N/A', and capacity reservation group. The 'Networking' tab shows the public IP address '13.87.135.146', private IP address '10.8.0.4', virtual network/subnet 'gr07CAP-vnet/default', and DNS name 'Configure'. Other tabs like 'Monitoring', 'Capabilities', 'Recommendations', and 'Tutorials' are also visible.

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
```

```

fran@docker07cap: ~$ docker inspect iblanque/python:20:07
[
  {
    "Id": "sha256:5b94763b174a1808ee91e51dc977ff8134ba38ddf1b4cdbac5e1278b1b9355",
    "RepoTags": [
      "iblanque/python:20:07"
    ],
    "Repository": "iblanque/python:20:07",
    "Parent": "",
    "Comment": "buildkit.dockerfile.v0",
    "Created": "2022-03-12T17:17:15.1195952Z",
    "Container": "",
    "ContainerConfig": {
      "Hostname": "",
      "Domainname": "",
      "User": "",
      "AttachStdin": false,
      "AttachStdout": false,
      "AttachStderr": false,
      "Tty": false,
      "OpenStdin": false,
      "StdinOnce": false,
      "Env": null,
      "Cmd": null,
      "Image": "",
      "Volumes": null,
      "WorkingDir": "",
      "Entrypoint": null,
      "OnBuild": null,
      "Labels": null
    },
    "DockerVersion": "",
    "Author": "",
    "Config": {
      "Hostname": "",
      "Domainname": "",
      "User": "",
      "AttachStdin": false,
      "AttachStdout": false,
      "AttachStderr": false,
      "Tty": false,
      "OpenStdin": false,
      "StdinOnce": false,
      "Env": [
        "PATH=/usr/local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin",
        "LANG=C.UTF-8",
        "PYTHONIOENCODING=UTF-8",
        "GPG_KEY=4D965ADCA6808637158AC36721BADD04FF",
        "PYTHON_VERSION=2.7.18",
        "PYTHON_PIP_VERSION=20.0.2",
        "PYTHON_GET_PIP_URL=https://github.com/pyupio/get-pip/raw/d5919733c69c378a222a28a3fa9ed3ee0a5d5/get-pip.py",
        "PYTHON_GET_PIP_SHA256=424cdd4c9c97708983776699749b1dcd4ea263809975071878cc189c"
      ],
      "Args": null,
      "ArgsEscaped": true,
      "Image": "",
      "Volumes": null,
      "WorkingDir": "",
      "Entrypoint": [
        "/bin/sh",

```

```
fran@docker07cap: ~  
"OpenStdin": false,  
"StdinOnce": false,  
"Env": [  
  "PATH=/usr/local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin",  
  "LANG=C.UTF-8",  
  "PYTHONIOENCODING=UTF-8",  
  "GPG_KEY=C01E1CAD5EA2C4F88E3371594C367C218ADD0FF",  
  "PYTHON_VERSION=2.7.18",  
  "PYTHON_PIP_VERSION=20.0.2",  
  "PYTHON_GET_PIP_URL=https://github.com/pypa/get-pip/raw/d59197a3c169cfe378a22428a3fa99d3e080a5d/get-pip.py",  
  "PYTHON_GET_PIP_SHA256=421ac1d4ccc9f930a88e337867d974b91bdceea26609275071878cc189e"  
],  
"Cmd": null,  
"ArgsEscaped": true,  
"Image": "",  
"Volumes": null,  
"WorkingDir": "",  
"Entrypoint": [  
  "/bin/sh",  
  "-c",  
  "/bin/encode01.py"  
],  
"OnBuild": null,  
"Labels": null  
},  
"Architecture": "amd64",  
"Os": "linux",  
"Size": 981717492,  
"VirtualSize": 981717492,  
"GraphDriver": {  
  "Data": {  
    "LowerDir": "/var/lib/docker/overlay2/21e7e157c63828845f9d5f7a121ab5b1f84f27463da8807c6b9808392a1/diff:/var/lib/docker/overlay2/5bda9ff15193f6cc7db9e856ae6e9ac628977d37814972c167a2d413ebf61e82/diff:/var/lib/docker/ove  
rlay2/619735d58f959f7659dc444ef654c5d21c8f2b1c34bd3d15397439bd220/diff:/var/lib/docker/overlay2/65ec94439793363e8a102021ce0f51ed48ff75208add286287f0bf1e73c9/diff:/var/lib/docker/overlay2/49e05bc987af8cda36409797f7d5e890f09cef  
06d20086ab36584f7570a/diff:/var/lib/docker/overlay2/9cfe34af0cd170667391eea62ebf7629014a2a9a64ecf915bd6d4d32d95be5ab/diff:/var/lib/docker/overlay2/dbff887e51bb69fead4315235f2a3896fba69ea61de668c841bcf7386264a67/diff:/var/lib/docker/ove  
rlay2/a19988cd6c4572659f7d2e080ff52effb8c8a540f929e0d870b1b8efcc3d1/diff:/var/lib/docker/overlay2/b15f9eb23b868d289537b8f5c5ff97bfa9e62fa3dcdfc20a730b494b6d00d7/diff",  
    "MergedDir": "/var/lib/docker/overlay2/57bdc1233bc610f18e418a6fa116ede07da42b499c907c2750153d45d78fa02d/merged",  
    "UpperDir": "/var/lib/docker/overlay2/57bdc1233bc610f18e418a6fa116ede07da42b499c907c2750153d45d78fa02d/diff",  
    "WorkDir": "/var/lib/docker/overlay2/57bdc1233bc610f18e418a6fa116ede07da42b499c907c2750153d45d78fa02d/work"  
  },  
  "Name": "overlay2"  
},  
"RootFS": {  
  "Type": "layers",  
  "Layers": [  
    "sha256:461710022993f933f943ca0e4ed09357c7d0b136c12bf4bd38a388854b",  
    "sha256:f1420c2af1ab82afc499a40f068206d0642560b1129effa7074a1b1fe07c1bf",  
    "sha256:a3c1026c6bcc2c6a8c352a9187e64939ebf37d964f3567ba530b019dc518e27d",  
    "sha256:d35c5bda4793162915162eccc03739566b72ff897f970ac675eaf75f55c29f5e",  
    "sha256:468293311a4829f0ebf7177b274b36377992290dc44e508d0b57a67a5ef9",  
    "sha256:47a58fb45d993a03d81c720b7ccf5d781414293e48cee5a7c4054ccc09d77ef",  
    "sha256:ceee881b0b968afda7299329cd51cf70f4979f0911ee25fca4adabaa710b",  
    "sha256:da7b8a804f221cb44069045925ab26e91123ede9cc78dd48c911fdbca2",  
    "sha256:e571d2d3c73cb5928867c3239c548918261b543a35db8fbfedf28cb75962644f",  
    "sha256:8a6efc91c085d9cc2121139a3a16d215675fc9cb68f83a908d91044d03d79"  
  ]  
},  
"Metadata": {  
  "LastTagTime": "0001-01-01T00:00:00Z"  
}  
}  
fran@docker07cap: $
```

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

b) Ejecutar el contenedor.

```
fran@docker07cap: ~  
fran@docker07cap:~$ docker run iblanque/python27:07  
U3VlcnRlIGVuIGVsIGV4YW1lbiAyMzQ=fran@docker07cap:~$
```

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:

```
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==>



A terminal window titled 'fran@msiFran:~' showing a curl command being executed. The command is: `curl "https://dec07cap.azurewebsites.net/api/decodefunc?code=90JAcVU2oUv6PeCRpj14aSUDjXKsyFF2VJXFntZfzRhkkGYI/awbRQ=="`. The output is: `b'Hola mundo'`. A green checkmark icon is visible in the bottom right corner of the terminal window.

