

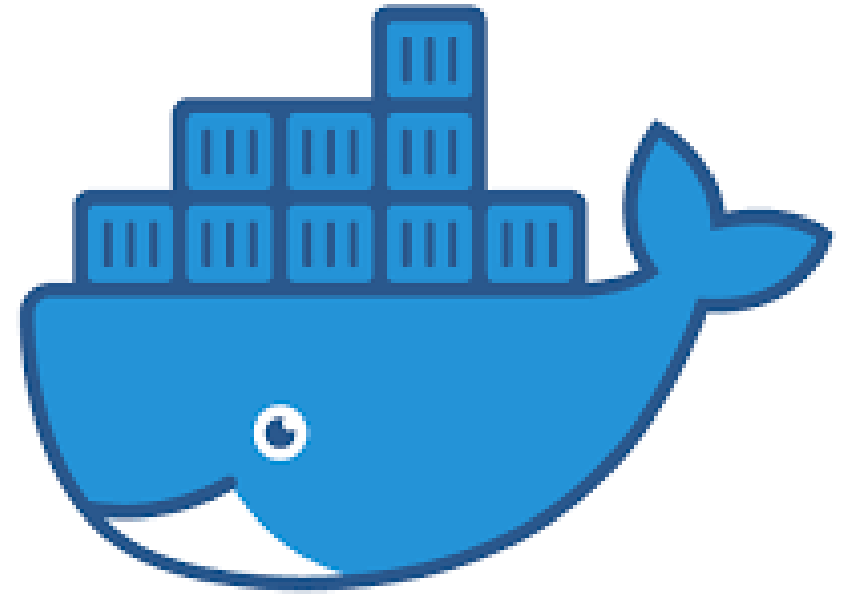


**Universidad
Andrés Bello®**
Conectar • Innovar • Liderar

ICF233 Ingeniería de Software II

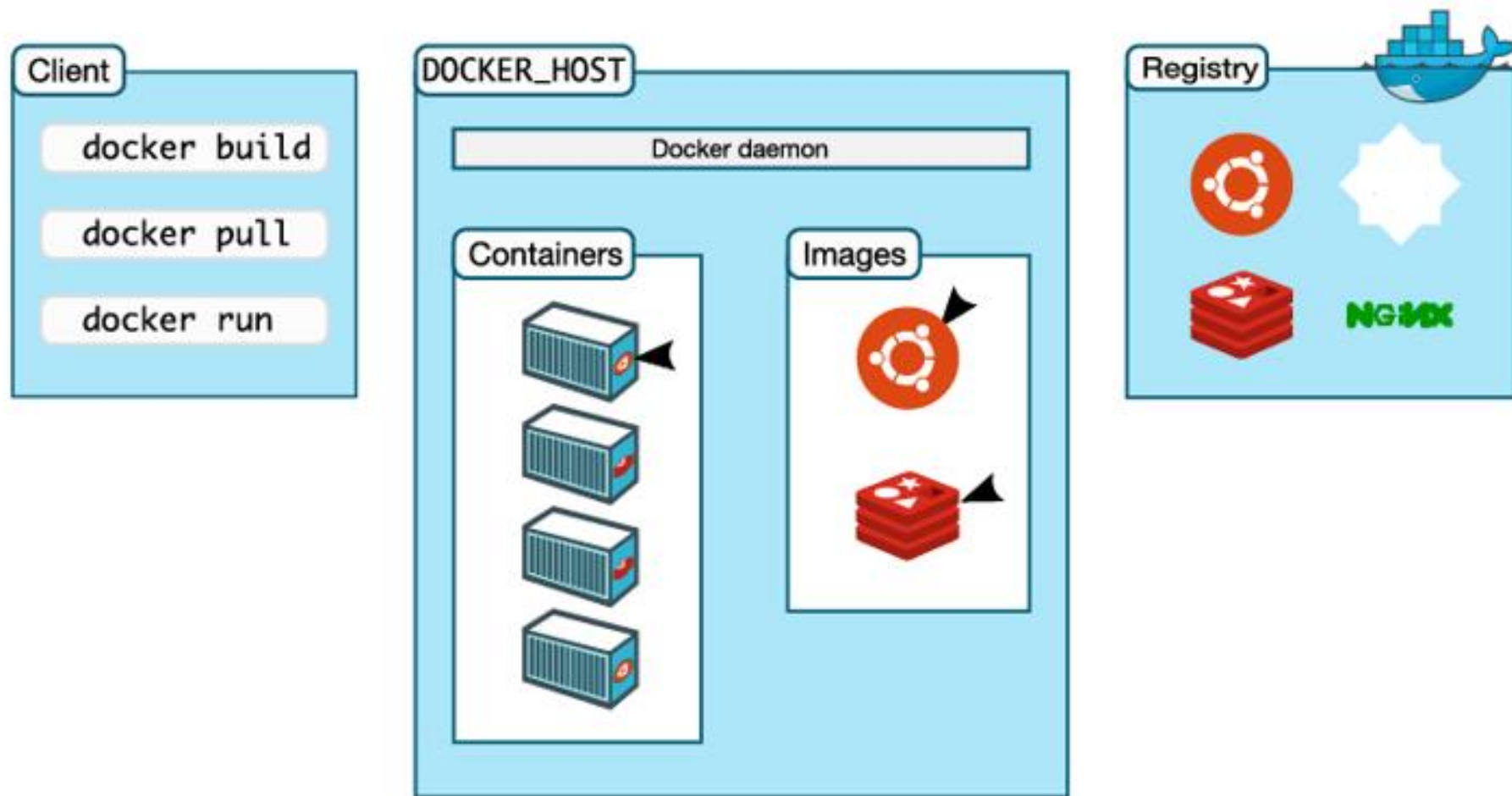
Segundo Semestre 2020

Docker

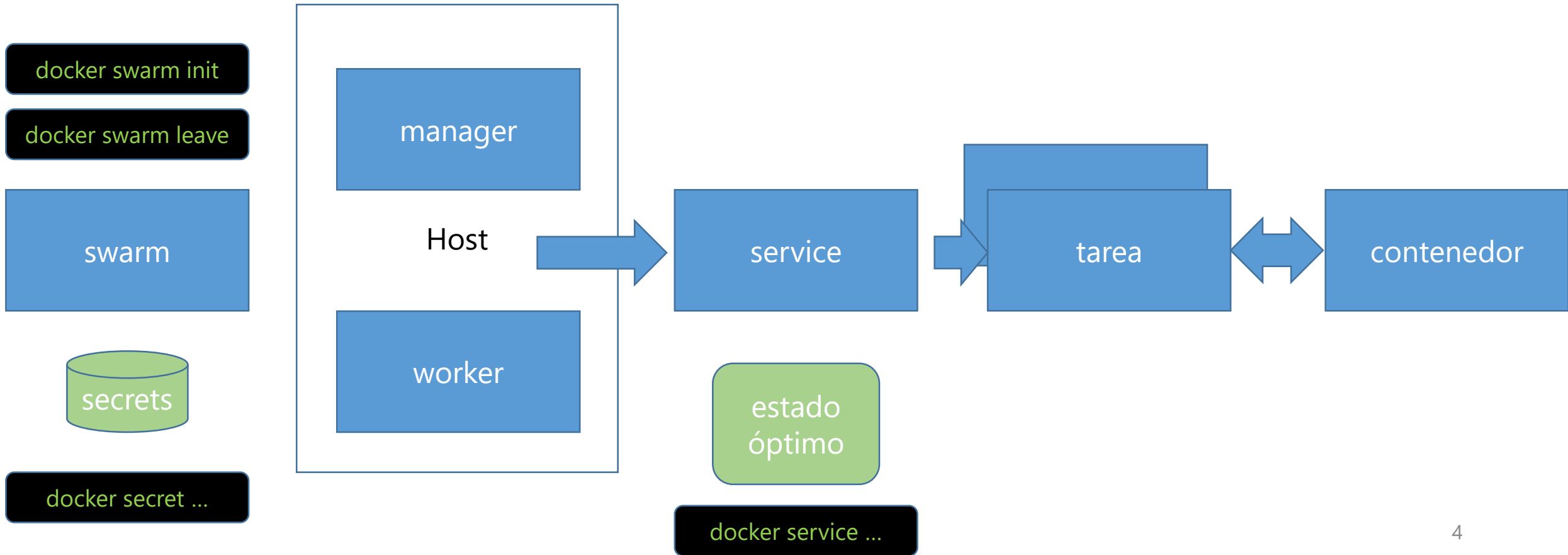


<https://hub.docker.com/editions/community/docker-ce-desktop-windows/>

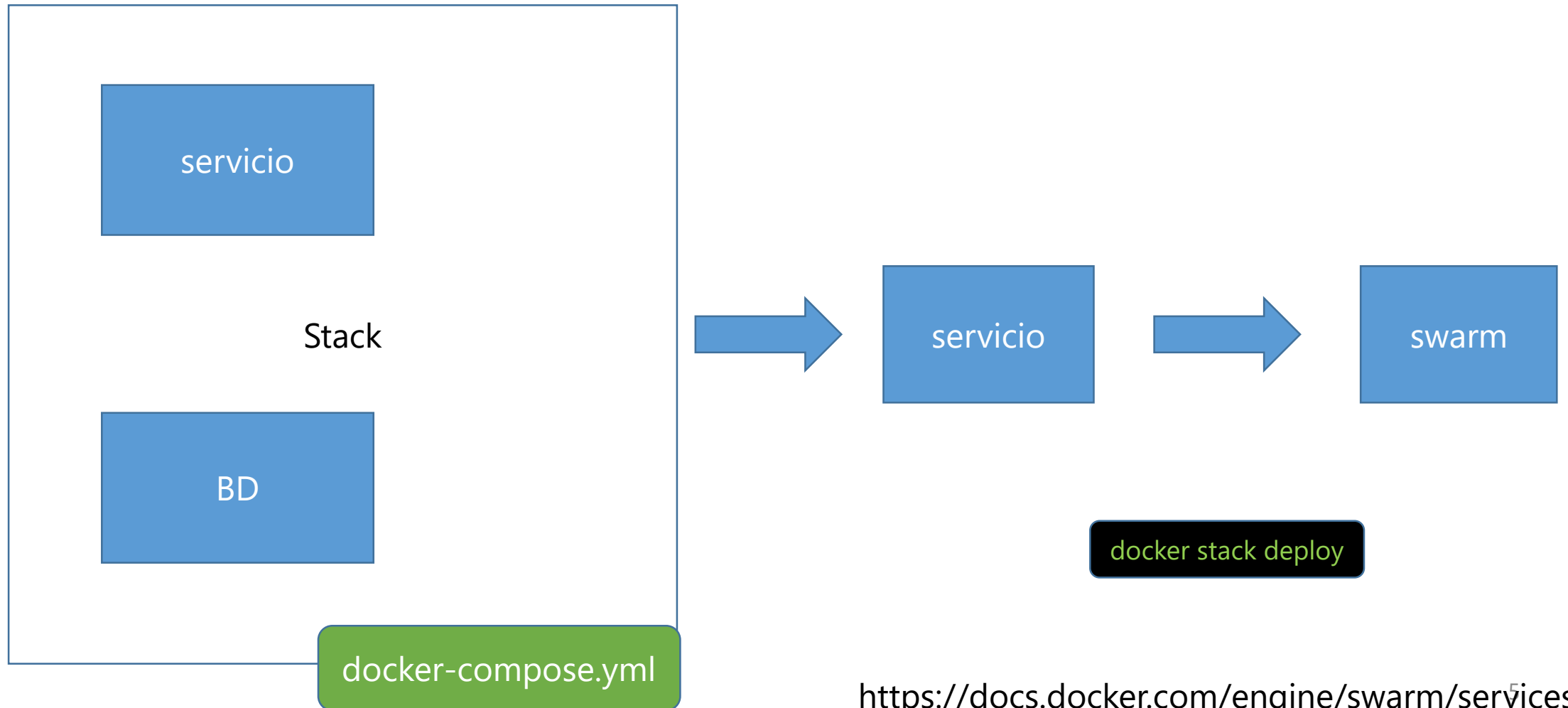
¿Qué es docker?



Docker: conceptos relevantes



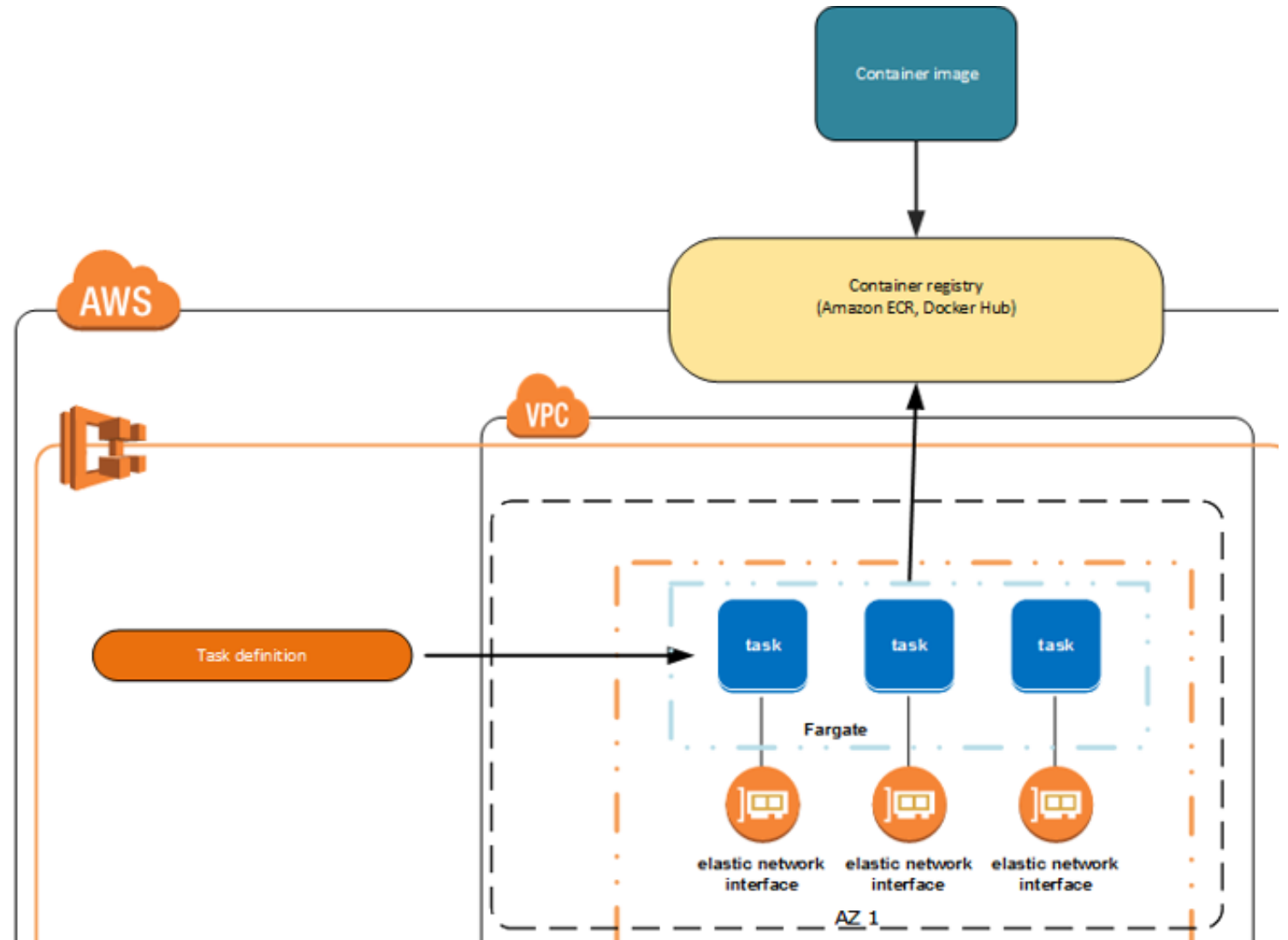
Docker conceptos relevantes



Amazon ECR/ECS y aws cli

<https://docs.aws.amazon.com/cli/latest/userguide/install-cliv2-windows.html>

aws --version



Obtener imagen base

```
C:\Users\pablo>docker run -it ubuntu
root@0eb0304b8724:/# ls -l
total 64
drwxr-xr-x  2 root root 4096 Mar  7 21:01 bin
drwxr-xr-x  2 root root 4096 Apr 24 2018 boot
drwxr-xr-x  5 root root  360 May  9 04:38 dev
drwxr-xr-x  1 root root 4096 May  9 04:38 etc
drwxr-xr-x  2 root root 4096 Apr 24 2018 home
drwxr-xr-x  8 root root 4096 May 23 2017 lib
drwxr-xr-x  2 root root 4096 Mar  7 21:00 lib64
drwxr-xr-x  2 root root 4096 Mar  7 21:00 media
drwxr-xr-x  2 root root 4096 Mar  7 21:00 mnt
drwxr-xr-x  2 root root 4096 Mar  7 21:00 opt
dr-xr-xr-x 142 root root    0 May  9 04:38 proc
drwx----- 2 root root 4096 Mar  7 21:01 root
drwxr-xr-x  1 root root 4096 Mar 12 00:20 run
drwxr-xr-x  1 root root 4096 Mar 12 00:20 sbin
drwxr-xr-x  2 root root 4096 Mar  7 21:00 srv
dr-xr-xr-x 13 root root    0 May  9 04:37 sys
drwxrwxrwt  2 root root 4096 Mar  7 21:01 tmp
drwxr-xr-x  1 root root 4096 Mar  7 21:00 usr
drwxr-xr-x  1 root root 4096 Mar  7 21:01 var
root@0eb0304b8724:/# cat /etc/lsb-release
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=18.04
DISTRIB_CODENAME=bionic
DISTRIB_DESCRIPTION="Ubuntu 18.04.2 LTS"
```

Entramos en modo interactivo
en la imagen
Para salir exit

Listar imágenes

```
C:\Users\pablo>docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
ubuntu	latest	94e814e2efa8	8 weeks ago	88.9MB
docker4w/nsenter-dockerd	latest	2f1c802f322f	6 months ago	187kB

Para borrar una imagen
docker rmi <imagen>
o
docker image rm

Containers en ejecución

```
C:\Users\pablo>docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
NAMES					

```
C:\Users\pablo>docker run -it --name "t3" ubuntu
```

```
root@11afdd2ad18b:/#
```

```
C:\Users\pablo>docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
NAMES					
11afdd2ad18b	ubuntu	"/bin/bash"	18 seconds ago	Up 16 seconds	
t3					

Para salir del container y dejarlo activo

CTRL P+Q

Para reingresar usar
docker attach

```
C:\Users\pablo>docker attach t3
```

```
root@11afdd2ad18b:/# ls -l
```

```
total 64
```

```
drwxr-xr-x  2 root root 4096 Mar  7 21:01 bin
```

```
drwxr-xr-x  2 root root 4096 Apr 24 2018 boot
```

```
drwxr-xr-x  5 root root  360 Jun  1 12:10 dev
```

```
drwxr-xr-x  1 root root 4096 Jun  1 12:10 etc
```

```
drwxr-xr-x  2 root root 4096 Apr 24 2018 home
```

Ver contenedores activos

```
C:\Users\pablo>docker container ls -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
11afdd2ad18b	ubuntu	"/bin/bash"	About a minute ago	Exited (0) 7 seconds ago	
	t3				

Para eliminar containers
detenidos
docker container prune

```
C:\Users\pablo>docker container prune
WARNING! This will remove all stopped containers.
Are you sure you want to continue? [y/N] y
Deleted Containers:
11afdd2ad18b5ba5c0512f7e4e2fe8949814825fca1db8a8df700a3adc0ae5b6

Total reclaimed space: 12B
```



MySQL en Docker

Obtenemos la imagen de mysql 5.7

En cada reinicio la base de datos se borra y no hay seguridad

```
C:\Users\pablo>docker run --name dbserver -e MYSQL_ROOT_PASSWORD=Secret.123 -p 3306:3306 -d mysql:5.7
Unable to find image 'mysql:5.7' locally
5.7: Pulling from library/mysql
743f2d6c1f65: Pull complete
3f0c413ee255: Pull complete
aef1ef8f1aac: Pull complete
f9ee573e34cb: Pull complete
3f237e01f153: Pull complete
f9da32e8682a: Pull complete
4b8da52fb357: Pull complete
6f38e9cfd49b: Pull complete
9f4834b3f44f: Pull complete
af631d92fdb8: Pull complete
0e771ddab25c: Pull complete
Digest: sha256:196fe3e00d68b2417a8cf13482bdab1fcc2b32cf7c7575d0906c700688b352b4
Status: Downloaded newer image for mysql:5.7
f1fec265a0284e859aa74528e0e4121e0ed855e4373532d282d0d9a31c6cc500
```

```
C:\Users\pablo>docker container ls
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
f1fec265a028	mysql:5.7	"docker-entrypoint.s..."	36 seconds ago	Up 32 seconds	0.0.0.0:3306->3306/tcp, 33060/tcp	dbserver

Para entrar en modo interactivo al container

```
C:\Users\pablo>docker exec -it f1fec265a028 bash
root@f1fec265a028:/# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.7.26 MySQL Community Server (GPL)

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```



Creando una imagen de MySQL

Primero construiremos manualmente la imagen

Crearemos nuestra propia imagen de mysql usando como base la imagen ubuntu

```
C:\Users\pablo>docker run -it --name mysqlserver ubuntu
root@752ca6d9c74d:/# apt-get update
Get:1 http://archive.ubuntu.com/ubuntu bionic InRelease [242 kB]
Get:2 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:3 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:4 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages [324 kB]
Get:5 http://archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:6 http://archive.ubuntu.com/ubuntu bionic/restricted amd64 Packages [13.5 kB]
Get:7 http://archive.ubuntu.com/ubuntu bionic/multiverse amd64 Packages [186 kB]
Get:8 http://archive.ubuntu.com/ubuntu bionic/main amd64 Packages [1344 kB]
Get:9 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 Packages [4168 B]
Get:10 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [486 kB]
Get:11 http://security.ubuntu.com/ubuntu bionic-security/restricted amd64 Packages [5436 B]
Get:12 http://archive.ubuntu.com/ubuntu bionic/universe amd64 Packages [11.3 MB]
Get:13 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [829 kB]
Get:14 http://archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [1215 kB]
Get:15 http://archive.ubuntu.com/ubuntu bionic-updates/restricted amd64 Packages [10.8 kB]
Get:16 http://archive.ubuntu.com/ubuntu bionic-updates/multiverse amd64 Packages [7236 B]
Get:17 http://archive.ubuntu.com/ubuntu bionic-backports/universe amd64 Packages [3902 B]
Get:18 http://archive.ubuntu.com/ubuntu bionic-backports/main amd64 Packages [2496 B]
Fetched 16.3 MB in 40s (405 kB/s)
Reading package lists... Done
root@752ca6d9c74d:/#
```

Instalamos mysql y nano

```
C:\Users\pablo>docker run -it --name mysqlserver ubuntu
root@752ca6d9c74d:/# apt-get update
Get:1 http://archive.ubuntu.com/ubuntu bionic InRelease [242 kB]
Get:2 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:3 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:4 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages [324 kB]
Get:5 http://archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:6 http://archive.ubuntu.com/ubuntu bionic/restricted amd64 Packages [13.5 kB]
Get:7 http://archive.ubuntu.com/ubuntu bionic/multiverse amd64 Packages [186 kB]
Get:8 http://archive.ubuntu.com/ubuntu bionic/main amd64 Packages [1344 kB]
```

```
root@752ca6d9c74d:/# apt-get -y install nano
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  spell
The following NEW packages will be installed:
  nano
0 upgraded, 1 newly installed, 0 to remove and 15 not upgraded.
Need to get 231 kB of archives.
After this operation, 778 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu bionic/main amd64 nano amd64 2.9.3-2 [231 kB]
Fetched 231 kB in 1s (179 kB/s)
```


Configuramos el servidor

```
root@752ca6d9c74d:/# cd /etc
root@752ca6d9c74d:/etc# cd mysql
root@752ca6d9c74d:/etc/mysql# cd conf.d
root@752ca6d9c74d:/etc/mysql/conf.d# ls
mysql.cnf  mysqldump.cnf
root@752ca6d9c74d:/etc/mysql/conf.d# nano docker.cnf
```

CTRL O
CTRL X
Para grabar y luego salir

root@752ca6d9c74d: /etc/mysql/conf.d

GNU nano 2.9.3 docker.cnf

```
[mysqld]
skip-host-cache
skip-name-resolve
```

^G Get Help	^O Write Out	^W Where Is	^K Cut Text	^J Justify	^C Cur Pos	M-U Undo
^X Exit	^R Read File	^_ Replace	^U Uncut Text	^T To Spell	^_ Go To Line	M-E Redo

Configuramos la seguridad de mysql

```
root@752ca6d9c74d:/etc/mysql/conf.d# nano docker.cnf
root@752ca6d9c74d:/etc/mysql/conf.d# cd
root@752ca6d9c74d:~# usermod -d /var/lib/mysql mysql
root@752ca6d9c74d:~# service mysql start
* Starting MySQL database server mysqld
root@752ca6d9c74d:~# mysql_secure_installation
```

[OK]

Securing the MySQL server deployment.

Connecting to MySQL using a blank password.

Y a todas las opciones
Nivel STRONG (3)
Escogemos una clave
Ej. Secret.123

Deshabilitamos el acceso anónimo de root

```
root@752ca6d9c74d:~# mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 5.7.26-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> alter user "root"@"localhost" identified with mysql_native_password by "Secret.123";
Query OK, 0 rows affected (0.00 sec)

mysql> exit
Bye
```

```
root@752ca6d9c74d:~# mysql
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: NO)
```

Creamos el usuario icf

```
root@752ca6d9c74d:~# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 5.7.26-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create user "icf"@"%" identified by "Secret.123";
Query OK, 0 rows affected (0.00 sec)

mysql> grant all privileges on *.* to "icf"@"%" identified by "Secret.123";
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)

mysql> exit
Bye
```

Configuramos el acceso remoto en mysqld.cnf comentando bind-address

CTRL O
CTRL X
Para grabar y
luego salir



root@752ca6d9c74d: ~

GNU nano 2.9.3 /etc/mysql/mysql.conf.d/mysqld.cnf

```
[mysqld_safe]
socket      = /var/run/mysqld/mysqld.sock
nice        = 0
```

```
[mysqld]
```

```
#
# * Basic Settings
#
```

```
user        = mysql
pid-file     = /var/run/mysqld/mysqld.pid
socket      = /var/run/mysqld/mysqld.sock
port        = 3306
basedir     = /usr
datadir     = /var/lib/mysql
tmpdir      = /tmp
lc-messages-dir = /usr/share/mysql
skip-external-locking
```

```
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
```

```
#bind-address = 127.0.0.1 127.0.0.1
```

```
#
```

^G Get Help

^O Write Out

^W Where Is

^K Cut Text

^J Justify

^C Cur Pos

M-U Undo

^X Exit

^R Read File

^_ Replace

^U Uncut Text

^T To Spell

^_ Go To Line

M-E Redo

Reiniciamos el servidor de base de datos, ajustamos las protecciones, lo detenemos y salimos

```
root@752ca6d9c74d:~# nano /etc/mysql/mysql.conf.d/mysqld.cnf
root@752ca6d9c74d:~# service mysql stop
* Stopping MySQL database server mysqld [ OK ]
root@752ca6d9c74d:~# chown -R mysql:mysql /var/lib/mysql
root@752ca6d9c74d:~# service mysql start
* Starting MySQL database server mysqld [ OK ]
root@752ca6d9c74d:~# service mysql stop
* Stopping MySQL database server mysqld [ OK ]
root@752ca6d9c74d:~# exit
exit
```

Hacemos el commit de la versión final de la imagen

```
C:\Users\pablo>docker container ls -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
752ca6d9c74d	ubuntu	"/bin/bash"	29 minutes ago	Exited (0) About a minute ago
	mysqlserver			

```
C:\Users\pablo>docker commit 752ca6d9c74d mysql
```

```
sha256:b08836e5be6554067e98c25c991ceb1043cdfc32af8c08ecbf7f2209dff2f3ac
```

```
C:\Users\pablo>docker image ls
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
mysql	latest	b08836e5be65	7 seconds ago	444MB

```
C:\Users\pablo>docker container prune
```

```
WARNING! This will remove all stopped containers.
```

```
Are you sure you want to continue? [y/N] y
```

```
Deleted Containers:
```

```
752ca6d9c74df797cdf5f9ee9ebe4bfba17ce92c2de15e6a7b2aa32a7614b161
```

```
Total reclaimed space: 354.8MB
```

Ahora tenemos nuestro servidor mysql en docker

```
chown -R mysql:mysql /var/lib/mysql && service mysql start
```

```
C:\Users\pablo>docker run -it --name dbserver -p 3306:3306 mysql
root@4ef238a9face:/# chown -R mysql:mysql /var/lib/mysql && service mysql start
```

```
* Starting MySQL database server mysqld
root@4ef238a9face:/#
```

[OK]

Para salir del contenedor y mantenerlo
activo
CTRL P+Q

```
root@4ef238a9face:/#
```

```
C:\Users\pablo>docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
4ef238a9face	mysql	"/bin/bash"	2 minutes ago	Up About a minute	0.0.0.0:3306->3306/tcp	dbserver

Nos conectamos a nuestro servidor desde el host

MySQL Workbench

Navigator: MySQL@127.0.0.1:3306 x

File Edit View Query Database Server Tools Scripting Help

Query 1 x

1 • select * from mysql.user;

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

SCHEMAS

Filter objects

sys

Result Grid

	Host	User	Select_priv	Insert_priv	Update_priv	Delete_priv
	localhost	root	Y	Y	Y	Y
	localhost	mysql.session	N	N	N	N
	localhost	mysql.svs	N	N	N	N
	localhost	debian-svs-maint	Y	Y	Y	Y
	%	icf	Y	Y	Y	Y
	NULL	NULL	NULL	NULL	NULL	NULL

Detener, reiniciar y volver a ingresar al contenedor y activar nuevamente el servidor

```
C:\Users\pablo>docker stop 4ef238a9face
4ef238a9face

C:\Users\pablo>docker start 4ef238a9face
4ef238a9face

C:\Users\pablo>docker attach 4ef238a9face
root@4ef238a9face:/# chown -R mysql:mysql /var/lib/mysql && service mysql start
* Starting MySQL database server mysqld
root@4ef238a9face:/# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.7.26-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

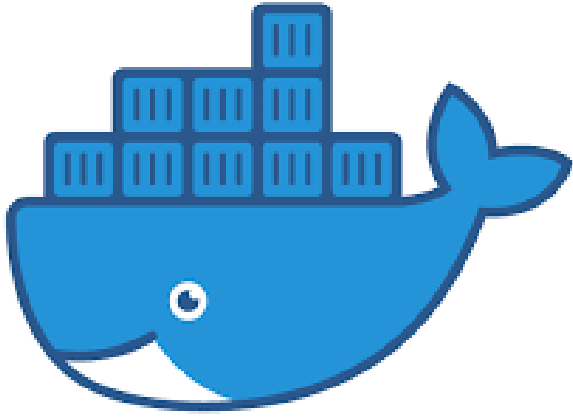
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> select user,host from mysql.user;
+-----+-----+
| user          | host          |
+-----+-----+
| icf           | %             |
| debian-sys-maint | localhost    |
| mysql.session | localhost    |
| mysql.sys     | localhost    |
| root          | localhost    |
+-----+-----+
5 rows in set (0.00 sec)
```

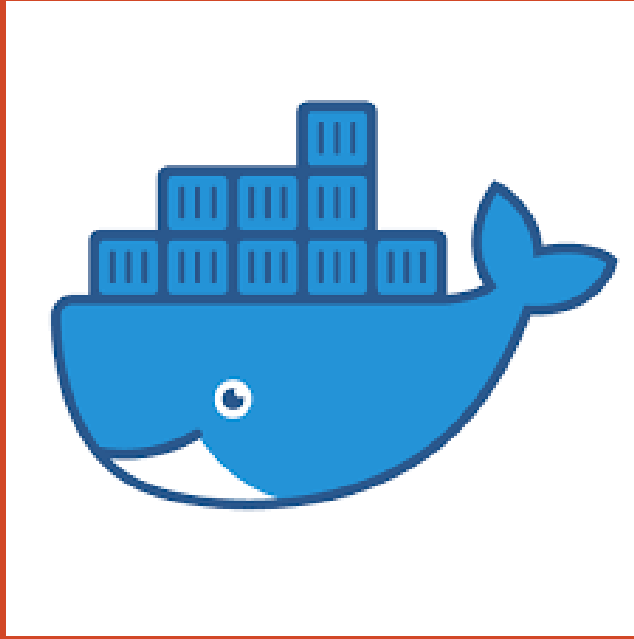
[OK]

Consideraciones

- Cada vez que realizamos un nuevo commit la imagen aumenta de tamaño, por lo que lo ideal es usar los menos pasos posibles
- El proceso de creación manual nos permite probar y depurar los pasos que nos permitirán automatizar la construcción usando un dockerfile
- Teniendo un dockerfile podemos usar docker build para construir la imagen



Dockerizando un servicio



Paso 1: Creamos una imagen de Mysql

Un Dockerfile simple y la shell asociada, start.sh

```
dockerfile x
1 FROM mysql
2
3 COPY start.sh /
4
5 ENTRYPOINT ["/start.sh"]
6
7 EXPOSE 3306/tcp
```

```
start.sh x
1 #!/bin/bash
2 chown -R mysql:mysql /var/lib/mysql && service mysql start && tail -f /dev/null
```

<https://docs.docker.com/engine/reference/builder/>

Construimos la imagen con docker build desde la carpeta en donde está el dockerfile

```
D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion\mysql>docker build . -t mysqlserver
Sending build context to Docker daemon 3.072kB
Step 1/4 : FROM mysql
---> b08836e5be65
Step 2/4 : COPY start.sh /
---> 637c43be308c
Step 3/4 : ENTRYPOINT ["/start.sh"]
---> Running in 7e95603f3cc9
Removing intermediate container 7e95603f3cc9
---> 1c6efcf987ea
Step 4/4 : EXPOSE 3306/tcp
---> Running in d7c38dc7c5a1
Removing intermediate container d7c38dc7c5a1
---> c9628a43407b
Successfully built c9628a43407b
Successfully tagged mysqlserver:latest
SECURITY WARNING: You are building a Docker image from Windows against a non-Windows Docker host. All files and directories added to build context
will have '-rwxr-xr-x' permissions. It is recommended to double check and reset permissions for sensitive files and directories.

D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion\mysql>docker image ls
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
mysqlserver	latest	c9628a43407b	24 seconds ago	444MB
mysql	latest	b08836e5be65	17 minutes ago	444MB

Creamos un contenedor usando la imagen en modo background, luego ejecutamos bash para ingresar

```
D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion\mysql>docker run -d --name dbserver -p 3306:3306 mysqlserver  
d1cc0e378f4526df7ddeed9b2cfd42f128d1cd868029377639e393728477e976
```

```
D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion\mysql>docker exec -it dbserver bash
```

```
root@d1cc0e378f45:/# ps -fea
```

UID	PID	PPID	C	STIME	TTY	TIME	CMD
root	1	0	0	13:50	?	00:00:00	/bin/bash /start.sh
root	6	1	1	13:50	?	00:00:00	chown -R mysql:mysql /var/lib/mysql
root	7	0	0	13:50	pts/0	00:00:00	bash
root	16	7	0	13:50	pts/0	00:00:00	ps -fea

```
root@d1cc0e378f45:/#
```

El contenedor ahora inicia el servidor en forma automática

Todo contenedor requiere un proceso que quede activo para mantenerse en ejecución de lo contrario terminaría

La base de datos está activa con las configuraciones asignadas

MySQL Workbench

Mysql@127.0.0.1:3306 x

File Edit View Query Database Server Tools Scripting Help

Navigator

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

SCHEMAS

Filter objects

sys

Query 1 x

```
select * from mysql.user;
```

Limit to 1000 rows

Result Grid

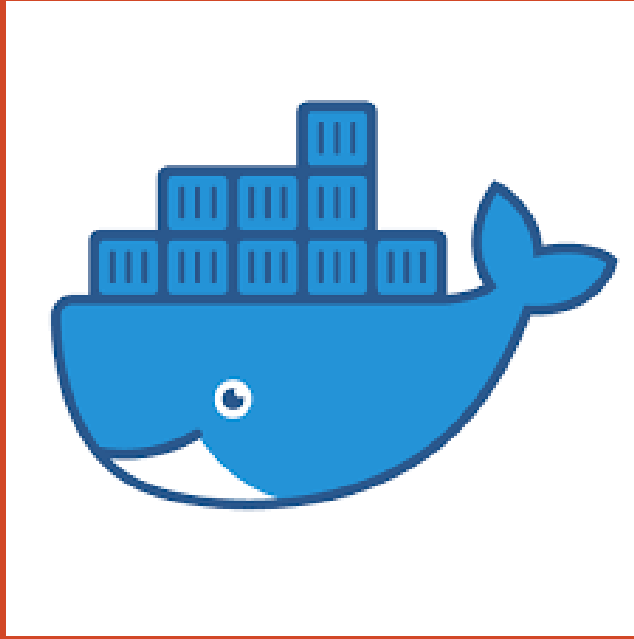
	Host	User	Select_priv	Insert_priv	Update_priv	Delete_priv	Create_priv	Drop_priv
	localhost	root	Y	Y	Y	Y	Y	Y
	localhost	mysql.session	N	N	N	N	N	N
	localhost	mysql.sys	N	N	N	N	N	N
	localhost	debian-sys-maint	Y	Y	Y	Y	Y	Y
	%	icf	Y	Y	Y	Y	Y	Y
	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Salimos del contenedor, lo detenemos y lo eliminamos

```
root@d1cc0e378f45:/# exit  
exit
```

```
D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion\mysql>docker stop dbserver  
dbserver
```

```
D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion\mysql>docker container rm dbserver  
dbserver
```



Paso 2: Conectando el servicio a Mysql

Creamos una carpeta mysql dentro de servicioplanificacion con el dockerfile y start.sh

└─ servicioplanificacion

├─ api

└─ mysql

├─ dockerfile

└─ start.sh

Crearemos una red para los servidores

Esto habilita un DNS entre las máquinas

```
C:\Users\pablo>docker network create --driver=bridge --subnet=192.168.0.0/24 br0  
c1f03e8d27a58dc5a116e3298cb1340fec9937ec69e49a75ef620e409bfff36b
```

```
C:\Users\pablo>docker network ls
```

NETWORK ID	NAME	DRIVER	SCOPE
c1f03e8d27a5	br0	bridge	local
a67e1013ad82	bridge	bridge	local
340f6dd84793	host	host	local
3c35cce5ac19	none	null	local

Ejecutamos el contenedor dentro de la red

```
docker run -d --name dbserver --network br0 -p 3306:3306 mysqlserver
```

```
docker exec -it dbserver bash
```

```
(venv) D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>docker run -d --name dbserver --network br0 -p 3306:3306 mysqlserver  
b44187d9e605326c3a757290c2d075820857756ca140f5a3fb5864e10c9cf3e2
```

```
(venv) D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>
```

```
(venv) D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>docker exec -it dbserver bash
```

```
root@b44187d9e605: /#
```

Instalamos ping para probar la conectividad

```
root@b44187d9e605:/# cat /etc/hosts
127.0.0.1    localhost
::1         localhost ip6-localhost ip6-loopback
fe00::0     ip6-localnet
ff00::0     ip6-mcastprefix
ff02::1     ip6-allnodes
ff02::2     ip6-allrouters
192.168.0.2  b44187d9e605
root@b44187d9e605:/# apt-get install iputils-ping
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libcap2 libcap2-bin libidn11 libpam-cap
The following NEW packages will be installed:
  iputils-ping libcap2 libcap2-bin libidn11 libpam-cap
0 upgraded, 5 newly installed, 0 to remove and 15 not upgraded.
Need to get 141 kB of archives.
After this operation, 537 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

```
root@b44187d9e605:/# ping dbserver
PING dbserver (192.168.0.2) 56(84) bytes of data.
64 bytes from b44187d9e605 (192.168.0.2): icmp_seq=1 ttl=64 time=0.054 ms
64 bytes from b44187d9e605 (192.168.0.2): icmp_seq=2 ttl=64 time=0.053 ms
^C
--- dbserver ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1050ms
rtt min/avg/max/mdev = 0.053/0.053/0.054/0.007 ms
root@b44187d9e605:/#
```

El DNS habilitado por
docker resuelve el
nombre del
contenedor

Creamos la base de datos planificacion y asignamos privilegios al usuario icf

```
root@b44187d9e605:/# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.7.26-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database planificacion character set utf8 collate utf8_bin;
Query OK, 1 row affected (0.00 sec)

mysql> grant all on planificacion.* to "icf"@"%" identified by "Secret.123";
Query OK, 0 rows affected, 1 warning (0.11 sec)

mysql> exit
Bye
root@b44187d9e605:/#
```


Verificamos el acceso

```
root@b44187d9e605:/# mysql -u icf -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 5
Server version: 5.7.26-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use planificacion;
Database changed
mysql> exit
Bye
root@b44187d9e605:/# exit
exit
```

Hacemos commit del contenedor para crear testdb

```
(venv) D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>docker exec -it dbserver bash
root@b44187d9e605:/# service mysql stop
* Stopping MySQL database server mysqld [ OK ]
root@b44187d9e605:/# exit
exit
```

```
(venv) D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>docker commit dbserver testdb
sha256:d855e69773604c8d8fa2514087007c1d5cc1acb4ea930515a89a405e075f59ab
```

```
(venv) D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>docker image ls
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
testdb	latest	d855e6977360	47 seconds ago	572MB
mysqlserver	latest	c9628a43407b	4 hours ago	444MB
mysql	latest	b08836e5be65	4 hours ago	444MB

Liberamos el espacio

```
(venv) D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>docker image prune
WARNING! This will remove all dangling images.
Are you sure you want to continue? [y/N] y
Total reclaimed space: 0B
```

```
(venv) D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>docker container stop dbserver
dbserver
```

```
(venv) D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>docker container prune
WARNING! This will remove all stopped containers.
Are you sure you want to continue? [y/N] y
Deleted Containers:
b44187d9e605326c3a757290c2d075820857756ca140f5a3fb5864e10c9cf3e2

Total reclaimed space: 128.5MB
```

Iniciamos el contenedor de base de datos a partir de testdb

```
docker run -d --name dbserver --network br0 -p 3306:3306 testdb
```

```
docker exec -it dbserver bash
```

```
(venv) D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>docker run -d --name dbserver --network br0 -p 3306:3306 testdb  
2bea17f9dda291aa1e68ea611ded6c1c814e2e6a7cc175fd16cc8d049c066781
```

```
(venv) D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
2bea17f9dda2	testdb	"/start.sh"	18 seconds ago	Up 16 seconds	0.0.0.0:3306->3306/tcp	dbserver

Agregamos conector a mysql y coverage a requirements.txt dentro de servicioplanificacion e instalamos

```
requirements.txt x
1  django
2  djangoRESTframework
3  gunicorn
4  mysqlclient
5  coverage
```

coverage nos permitirá
ver la cobertura de las
pruebas unitarias más
adelante

```
(venv) D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>pip3 install -r requirements.txt
```

```
Collecting mysqlclient
  Downloading https://files.pythonhosted.org/packages/e9/30/b40f39fcc6c39038cad3263b3b4280133fb033e0565fe4127d507aa011fe/mysqlclient-1.4.2.post1-cp37-cp37m-win_amd64.whl (178kB)
    |████████████████████| 184kB 2.2MB/s
Installing collected packages: mysqlclient
Successfully installed mysqlclient-1.4.2.post1
```

Conectamos el servicio REST a Mysql editando settings.py

```
DATABASES = {  
    # 'default': {  
    #     'ENGINE': 'django.db.backends.sqlite3',  
    #     'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),  
    # },  
    'default': {  
        'ENGINE': 'django.db.backends.mysql',  
        'NAME': 'planificacion',  
        'USER': 'icf',  
        'PASSWORD': 'Secret.123',  
        'DEFAULT-CHARACTER-SET': 'utf8',  
        'HOST': '127.0.0.1',  
        'PORT': '3306',  
        'TEST': {  
            'NAME': 'planificacion_test',  
        }  
    },  
}
```

Creamos tres archivos de settings para los ambientes y la carpeta static en la raíz del componente

```
utest.py x ... test.py x ... production.py x
1 from .settings import *
2
3 DATABASES = {
4     'default': {
5         'ENGINE': 'django.db.backends.sqlite3',
6         'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
7     },
8 }

1 from .settings import *
2
3 DATABASES = {
4     'default': {
5         'ENGINE': 'django.db.backends.mysql',
6         'NAME': 'planificacion',
7         'USER': 'icf',
8         'PASSWORD': 'Secret.123',
9         'DEFAULT-CHARACTER-SET': 'utf8',
10        'HOST': '127.0.0.1',
11        'PORT': '3306',
12        'TEST': {
13            'NAME': 'planificacion_test',
14        },
15    },
16 }

1 from .settings import *
2
3 DATABASES = {
4     'default': {
5         'ENGINE': 'django.db.backends.mysql',
6         'NAME': 'planificacion',
7         'USER': 'icf',
8         'PASSWORD': 'Secret.123',
9         'DEFAULT-CHARACTER-SET': 'utf8',
10        'HOST': '192.168.0.2',
11        'PORT': '3306',
12        'TEST': {
13            'NAME': 'planificacion_test',
14        },
15    },
16 }
17
18 STATIC_ROOT = '/servicioplanificacion/static'
19
```

<https://docs.djangoproject.com/en/2.1/howto/static-files/>

Podemos usar sqlite para pruebas locales rápidas con los test unitarios

```
(venv) D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>set DJANGO_SETTINGS_MODULE=servicioplanificacion.utest
(venv) D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
June 09, 2019 - 11:33:29
Django version 2.2.1, using settings 'servicioplanificacion.utest'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```


Usamos MySQL para el ambiente de test usando el contenedor de mysql

```
(venv) D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>set DJANGO_SETTINGS_MODULE=servicioplanificacion.test
(venv) D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
```

Usamos MySQL para el ambiente de produccion usando el contenedor de mysql con la red br0

```
(venv) D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>set DJANGO_SETTINGS_MODULE=servicioplanificacion.production
(venv) D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
```

Ejecutamos las migraciones usando las settings de test

```
(venv) D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, api, auth, authtoken, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying api.0001_initial... OK
  Applying api.0002_auto_20190507_2306... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying authtoken.0001_initial... OK
  Applying authtoken.0002_auto_20160226_1747... OK
  Applying sessions.0001_initial... OK
```

Creamos un superusuario

```
(venv) D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>python manage.py createsuperuser
Username (leave blank to use 'pablo'): servidor
Email address:
Password:
Password (again):
Superuser created successfully.
```

Podemos hacer un nuevo commit si deseamos que la imagen tenga el superusuario disponible en testdb, y algunos datos de prueba.

De lo contrario la base de datos comienza en blanco y las pruebas deben cargar los datos

Hacemos commit de testdb

```
(venv) D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>docker exec -it dbserver bash
```

```
root@2bea17f9dda2:/# service mysql stop
```

```
* Stopping MySQL database server mysqld
```

[OK]

```
root@2bea17f9dda2:/# exit
```

```
exit
```

```
(venv) D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>docker container commit dbserver testdb
```

```
sha256:7d75c9f8750b9d82fae8632fafc027072f18f065f48eeaead2eb94261b32dbd6
```

```
(venv) D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>docker image ls
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
testdb	latest	7d75c9f8750b	6 seconds ago	700MB
mysqlserver	latest	c9628a43407b	4 hours ago	444MB
mysql	latest	b08836e5be65	5 hours ago	444MB

Reiniciamos el servidor

```
(venv) D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
2bea17f9dda2	d855e6977360	"/start.sh"	11 minutes ago	Up 11 minutes	0.0.0.0:3306->3306/tcp	dbserver

```
(venv) D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>docker container stop dbserver
dbserver
```

```
(venv) D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>docker container prune
```

WARNING! This will remove all stopped containers.

Are you sure you want to continue? [y/N] y

Deleted Containers:

2bea17f9dda291aa1e68ea611ded6c1c814e2e6a7cc175fd16cc8d049c066781

Total reclaimed space: 128.2MB

```
docker run -d --name dbserver --network br0 -p 3306:3306 testdb
```

```
(venv) D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>docker run -d --name dbserver --network br0 -p 3306:3306 testdb
0a631b3d3b5dfdf0678fa8bf04a437a380642b66dd1df7efc6ed165017a2ffb
```

Actualizamos tests.py para usar la seguridad

```
from django.test import Client
from django.contrib.auth.models import User
```

```
class ViewTestCase(TestCase):
    """Esta clase define la testsuite para la API REST."""
    def setUp(self):
        """Definición de variables generales"""
        self.user=User.objects.create_superuser('servidor', 'api@test.com', "Secret.123")
        c = Client()
        response = c.post('/api/login/', {'username': 'servidor', 'password': 'Secret.123'})
        self.token = response.json()["token"]
        self.client = APIClient()
        self.client.credentials(HTTP_AUTHORIZATION='Token ' + self.token)

        curso_data = {'sigla': 'ICF121','nombre':'Introducción a la Ingeniería Civil Informática', 'creditos':6}
        self.response_setup = self.client.post(
            reverse('create'),
            curso_data,
            format="json")
```

<https://docs.djangoproject.com/en/2.2/topics/testing/tools/>

<https://docs.djangoproject.com/en/2.2/ref/request-response/>

<https://www.django-rest-framework.org/api-guide/testing/>

Ejecutamos los test usando la nueva base de datos

```
(venv) D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>python manage.py test -v 2
Creating test database for alias 'default' ('planificacion_test')...
Operations to perform:
  Synchronize unmigrated apps: messages, rest_framework, staticfiles
  Apply all migrations: admin, api, auth, authtoken, contenttypes, sessions
Synchronizing apps without migrations:
  Creating tables...
  Running deferred SQL...
Running migrations:
  Applying contenttypes.0001_initial... OK

test_creacion_de_curso (api.tests.CursoTestCase)
Test de creación de un curso ... ok
test_api_actualizar_curso (api.tests.ViewTestCase)
Test de actualización de curso a través de la API. ... ok
test_api_borrar_curso (api.tests.ViewTestCase)
Test borrado de curso a través de la API. ... ok
test_api_creacion_de_cursos (api.tests.ViewTestCase)
Test creación de curso a través de la API. ... ok
test_api_obtener_curso (api.tests.ViewTestCase)
Test de obtención de curso a través de la API. ... ok

-----
Ran 5 tests in 5.558s

OK
Destroying test database for alias 'default' ('planificacion_test')...
```

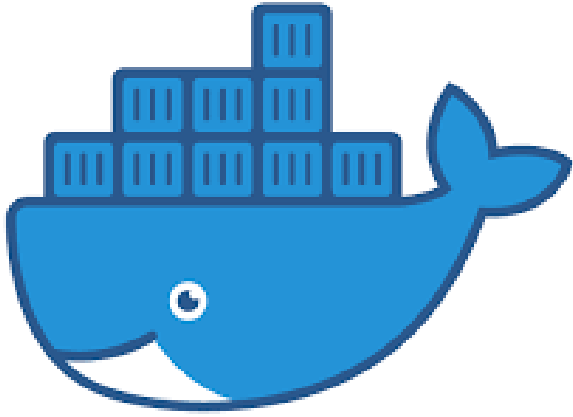

Ejecutamos las pruebas con cálculo de cobertura

```
(venv) D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>coverage run --source="." manage.py test api
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
.....
-----
Ran 5 tests in 4.012s

OK
Destroying test database for alias 'default'...
```

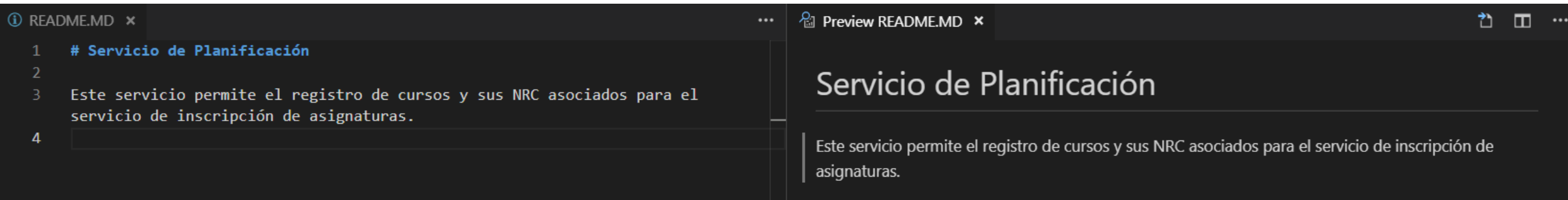
Obtenemos el informe de cobertura

```
(venv) D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>coverage report
Name                               Stmts  Miss  Cover
-----
api\__init__.py                     0      0   100%
api\admin.py                        1      0   100%
api\apps.py                         3      3     0%
api\migrations\0001_initial.py      5      0   100%
api\migrations\0002_auto_20190507_2306.py 4      0   100%
api\migrations\__init__.py          0      0   100%
api\models.py                       7      0   100%
api\serializers.py                  7      0   100%
api\tests.py                        44      0   100%
api\urls.py                         7      0   100%
api\views.py                        31      2    94%
manage.py                           9      2    78%
servicioplanificacion\__init__.py    0      0   100%
servicioplanificacion\production.py  2      2     0%
servicioplanificacion\settings.py   19      0   100%
servicioplanificacion\test.py        2      0   100%
servicioplanificacion\urls.py        5      0   100%
servicioplanificacion\utest.py       2      2     0%
servicioplanificacion\wsgi.py        4      4     0%
-----
TOTAL                               152     15    90%
```



Paso 2: Preparamos el componente

Creamos README.MD



Creamos .gitignore

```
# Byte-compiled / optimized / DLL files
__pycache__/
*.py[od]
*$py.class

# C extensions
*.so

# Distribution / packaging
.Python
build/
develop-eggs/
dist/
downloads/
eggs/
.eggs/
lib/
lib64/
parts/
sdist/
var/
wheels/
*.egg-info/
.installed.cfg
*.egg
MANIFEST
```

```
# Unit test / coverage reports
htmlcov/
.tox/
.coverage
.coverage.*
.cache
nosetests.xml
coverage.xml
*.cover
.hypothesis/
.pytest_cache/

# Django stuff:
*.log
local_settings.py
db.sqlite3

# Environments
.env
.venv
env/
venv/
ENV/
env.bak/
venv.bak/
```

<https://github.com/github/gitignore/blob/master/Python.gitignore>

En el directorio del proyecto iniciamos el repositorio

```
(venv) D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>git init
Initialized empty Git repository in D:/Clases/UNAB/ICF232-201910/servicios/servicioplanificacion/.git/

(venv) D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>git add .
warning: LF will be replaced by CRLF in .gitignore.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in mysql/start.sh.
The file will have its original line endings in your working directory.

(venv) D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>git config user.name "pabloschwarzenberg"

(venv) D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>git config user.email "pablo.schwarzenberg@unab.cl"

(venv) D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>git commit -m "inicio"
[master (root-commit) 9c3d4de] inicio
25 files changed, 609 insertions(+)
create mode 100644 .gitignore
create mode 100644 README.MD
create mode 100644 api/__init__.py
```

Creamos el proyecto en GitHub y conectamos el proyecto

<https://github.com/pabloschwarzenberg/ICF233-servicio>

```
D:\Clases\UNAB\201910\ICF232-201910\servicios\servicioplanificacion>git remote add origin https://github.com/pabloschwarzenberg/ICF233-servicio.git
D:\Clases\UNAB\201910\ICF232-201910\servicios\servicioplanificacion>git push -u origin master
fatal: HttpRequestException encountered.
  An error occurred while sending the request.
Username for 'https://github.com': pabloschwarzenberg
Password for 'https://pabloschwarzenberg@github.com':
Counting objects: 105, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (100/100), done.
Writing objects: 100% (105/105), 15.29 KiB | 355.00 KiB/s, done.
```

El proyecto en github

pabloschwarzenberg / ICF233-servicio

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Security Insights Settings

servicio web django para app-movil [Edit](#)

[Manage topics](#)

18 commits 1 branch 0 releases 0 contributors

Branch: master New pull request Create new file Upload files Find file Clone or download

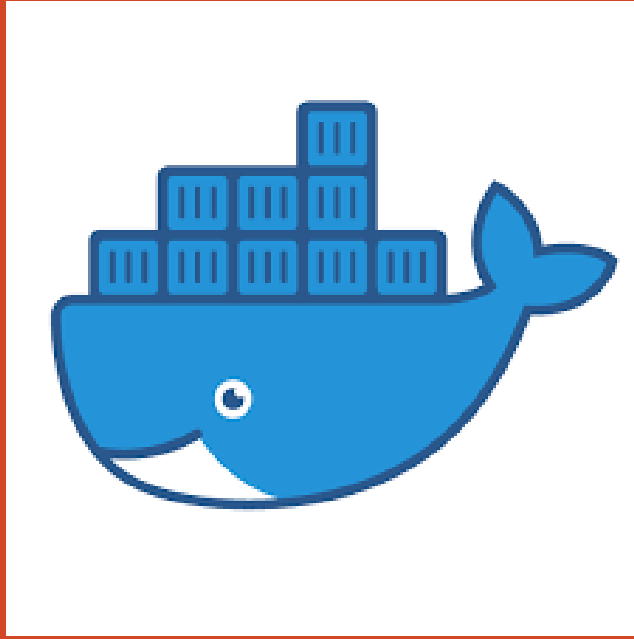
pabloschwarzenberg cors Latest commit abd98ea on Jun 13

api	inicio	5 months ago
mysql	inicio	5 months ago
servicioplanificacion	cors	5 months ago
.dockerignore	docker	5 months ago
.gitignore	inicio	5 months ago
Procfile	heroku	5 months ago
README.MD	inicio	5 months ago
docker-compose.yml	produccion	5 months ago
dockerfile	docker	5 months ago
heroku_secret.py	heroku	5 months ago
manage.py	inicio	5 months ago
requirements.txt	cors	5 months ago
secret.py	produccion	5 months ago
servicio.conf	produccion	5 months ago
setup.sh	produccion	5 months ago
start.sh	produccion	5 months ago

README.MD

Servicio de Planificación

Este servicio permite el registro de cursos y sus NRC asociados para el servicio de inscripción de asignaturas.



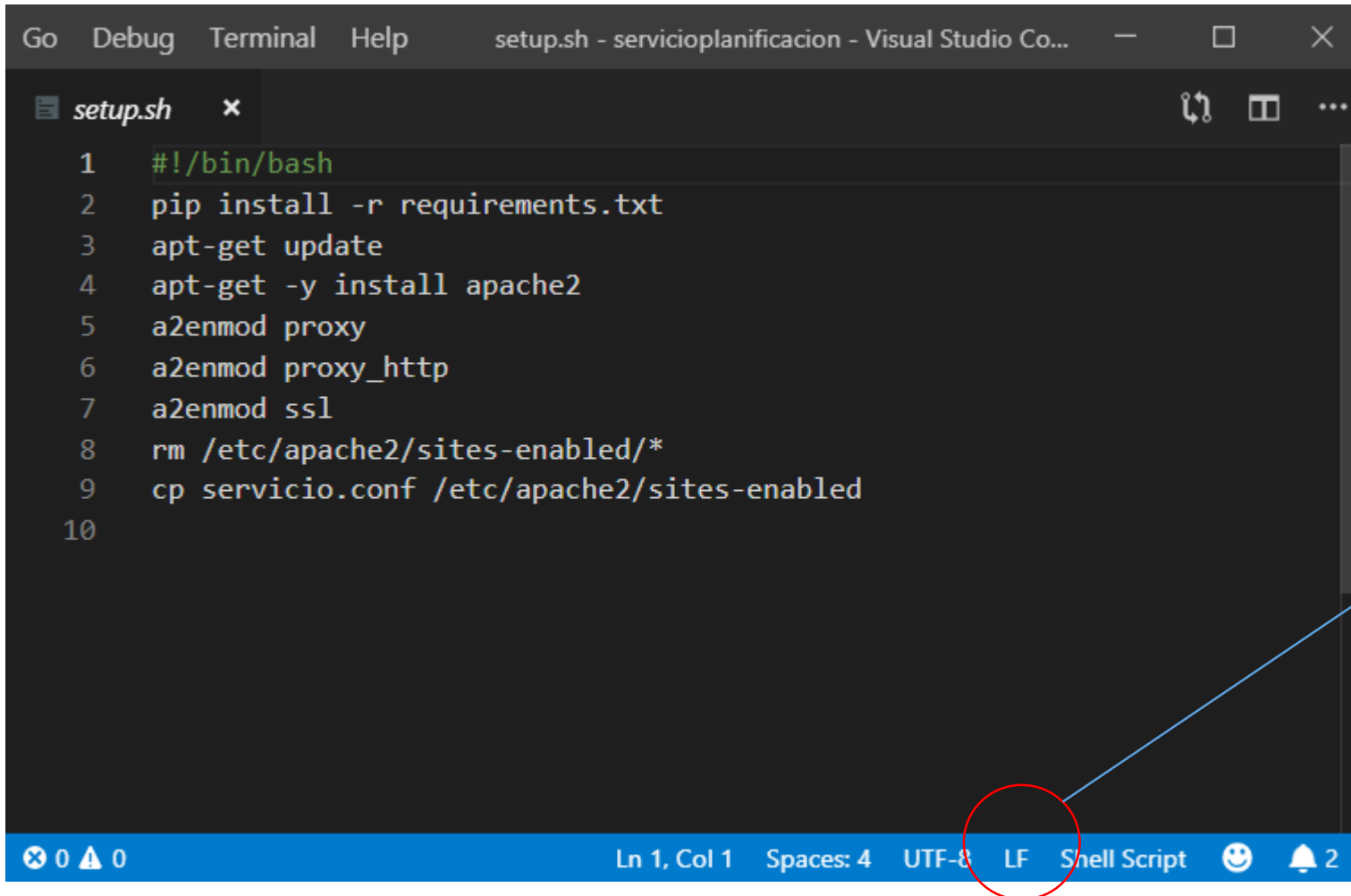
Paso 3: Preparamos el servicio

Creamos .dockerignore igual a .gitignore y agregamos .git en la primera línea



```
.dockerignore x
1  .git
2
3  # Byte-compiled / optimized / DLL files
4  __pycache__/
5  *.py[cod]
6  *$py.class
7
8  # C extensions
9  *.so
10
11 # Distribution / packaging
12 .Python
13 build/
14 develop-eggs/
15 dist/
16 downloads/
17 eggs/
18 .eggs/
19 lib/
```

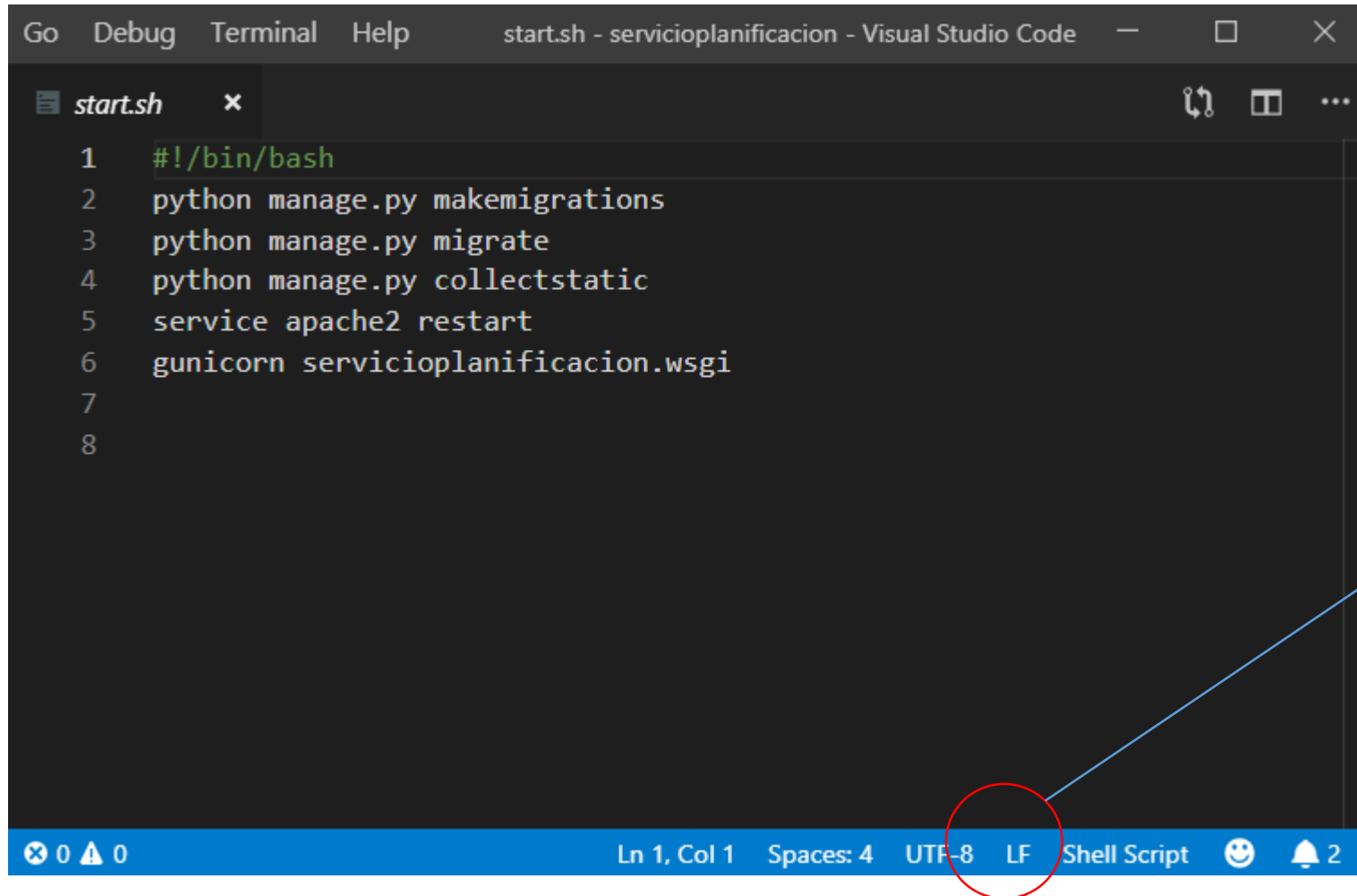
Creamos la shell setup.sh en la raíz del proyecto (formato UNIX)



```
Go Debug Terminal Help setup.sh - servicioplanificacion - Visual Studio Co...  
setup.sh x  
1 #!/bin/bash  
2 pip install -r requirements.txt  
3 apt-get update  
4 apt-get -y install apache2  
5 a2enmod proxy  
6 a2enmod proxy_http  
7 a2enmod ssl  
8 rm /etc/apache2/sites-enabled/*  
9 cp servicio.conf /etc/apache2/sites-enabled  
10  
Ln 1, Col 1 Spaces: 4 UTF-8 LF Shell Script 2
```

Al hacer clic en el
terminador de línea
podemos definir LF

Creamos la shell start.sh en la raíz del proyecto (formato UNIX)



The screenshot shows the Visual Studio Code editor with a file named `start.sh` open. The file contains the following content:

```
1  #!/bin/bash
2  python manage.py makemigrations
3  python manage.py migrate
4  python manage.py collectstatic
5  service apache2 restart
6  gunicorn servicioplanificacion.wsgi
7
8
```

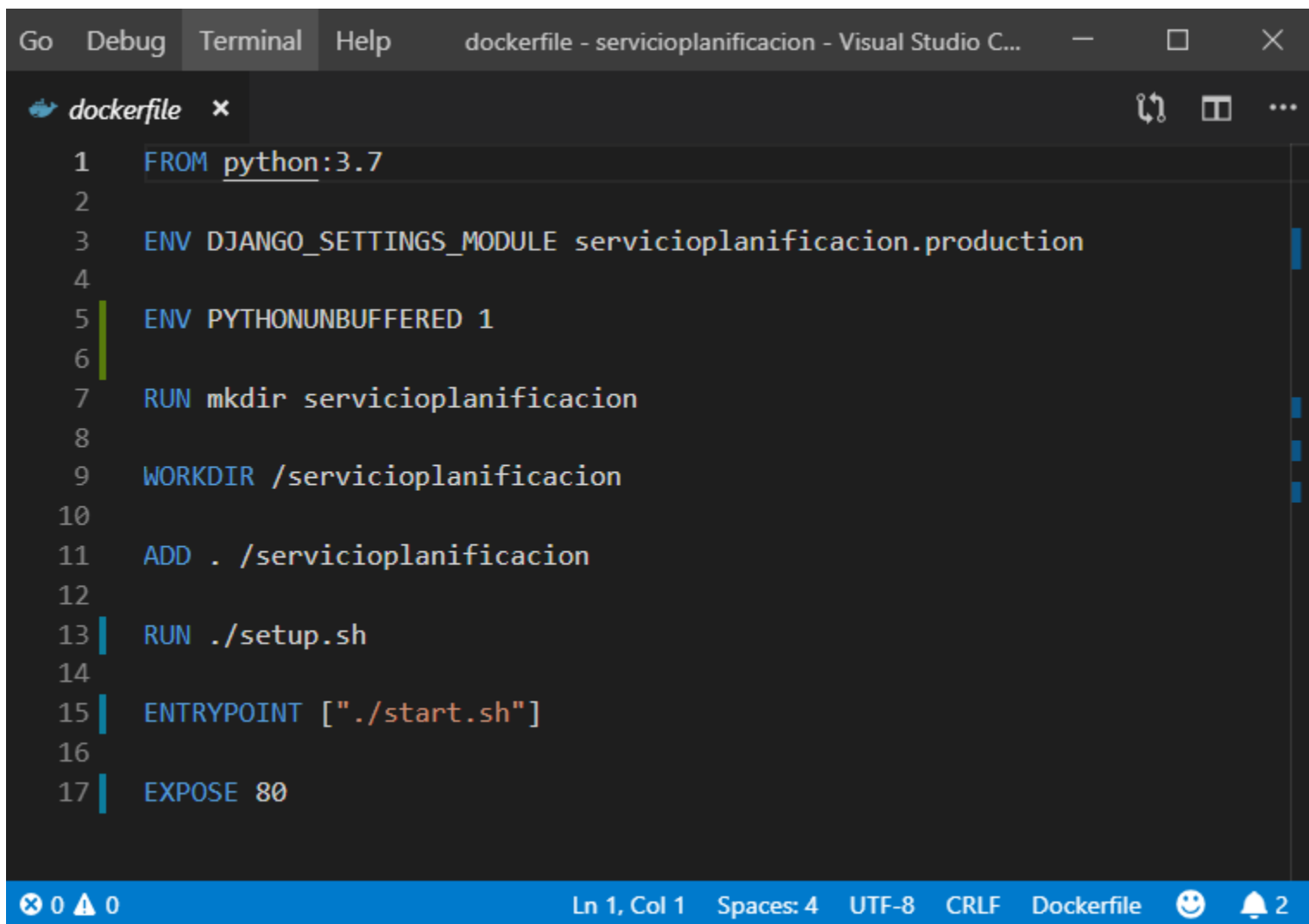
The status bar at the bottom indicates the file is a Shell Script, using UTF-8 encoding, with LF line endings. The 'LF' text in the status bar is circled in red, with a blue arrow pointing from a yellow callout box to it.

Al hacer clic en el
terminador de línea
podemos definir LF

Creamos servicio.conf para apache

```
servicio.conf x
1 <VirtualHost *:80>
2     DocumentRoot "/servicioplanificacion"
3     LogLevel error ssl:warn
4
5     ErrorLog ${APACHE_LOG_DIR}/servicio-error.log
6     CustomLog ${APACHE_LOG_DIR}/servicio-access.log combined
7
8     Alias /static /servicioplanificacion/static
9     <Directory "/servicioplanificacion/static">
10         Order Allow,Deny
11         Allow from All
12         Require all granted
13     </Directory>
14
15     ProxyPass /static !
16     ProxyPass / http://localhost:8000/
17     ProxyPassReverse / http://localhost:8000/
18 </VirtualHost>
```

Creamos el dockerfile en la raíz del componente



The image shows a Visual Studio Code editor window with a file named 'dockerfile' open. The editor has a dark theme and a sidebar on the left showing the file explorer. The Dockerfile content is as follows:

```
1 FROM python:3.7
2
3 ENV DJANGO_SETTINGS_MODULE servicioplanificacion.production
4
5 ENV PYTHONUNBUFFERED 1
6
7 RUN mkdir servicioplanificacion
8
9 WORKDIR /servicioplanificacion
10
11 ADD . /servicioplanificacion
12
13 RUN ./setup.sh
14
15 ENTRYPOINT ["./start.sh"]
16
17 EXPOSE 80
```

The status bar at the bottom indicates the current line and column (Ln 1, Col 1), the number of spaces (Spaces: 4), the encoding (UTF-8), the line ending (CRLF), the file type (Dockerfile), and the number of errors (0) and warnings (0).

Construimos la imagen

```
D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>docker build . -t servicio
Sending build context to Docker daemon 73.73kB
Step 1/9 : FROM python:3.7
----> a4cc999cf2aa
Step 2/9 : ENV DJANGO_SETTINGS_MODULE servicioplanificacion.production
----> Using cache
----> 754c94907e14
Step 3/9 : ENV PYTHONUNBUFFERED 1
----> Using cache
----> 6b1a4bc80fb4
Step 4/9 : RUN mkdir servicioplanificacion
----> Using cache
----> d1720ca98dda
Step 5/9 : WORKDIR /servicioplanificacion
----> Using cache
```

Iniciamos el container del servicio en la misma red

```
D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>docker run -d --name server --network br0 -p 80:80 servicio
45dae628609148912e5e2feef7b736b51bdf0e7266500d131965c681b5821c
```

```
D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
45dae6286091	servicio	"/start.sh"	8 seconds ago	Up 6 seconds	0.0.0.0:80->80/tcp	server
0a631b3d3b5d	testdb	"/start.sh"	5 hours ago	Up 5 hours	0.0.0.0:3306->3306/tcp	dbserver

```
D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>docker logs server
```

No changes detected

Operations to perform:

Apply all migrations: admin, api, auth, authtoken, contenttypes, sessions

Running migrations:

No migrations to apply.

152 static files copied to '/servicioplanificacion/static'.

AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 192.168.0.3. Set the 'ServerName' directive globally to suppress this message

Restarting Apache httpd web server: apache2.

[2019-06-09 22:53:09 +0000] [52] [INFO] Starting gunicorn 19.9.0

[2019-06-09 22:53:09 +0000] [52] [INFO] Listening at: http://127.0.0.1:8000 (52)

[2019-06-09 22:53:09 +0000] [52] [INFO] Using worker: sync

[2019-06-09 22:53:09 +0000] [95] [INFO] Booting worker with pid: 95

docker ps muestra los contenedores en ejecución
docker logs muestra el log de un contenedor

Probamos el servicio

```
curl -X POST -F "username=servidor" -F "password=Secret.123" http://localhost/api/login
```

```
D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>curl -X POST -F "username=servidor" -F "password=Secret.123" http://localhost/api/login
{"token":"0e35f50dd34a2a10b4d9386f2196d526cc567721"}
```

localhost/curso/

Unit tests | Django... mysql | Docker Doc... mysql - Docker Hub Cómo regar la man... Cultivo de plantas... docker commit | Do... Tutorial: Guía de ini... DIY WiFi RGB LED... Cybersecurity Fram... Tutorial de D

Django REST framework servidor

Create

Create OPTIONS GET

Vista que representa el comportamiento de la API REST.

GET /curso/

HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept
[]

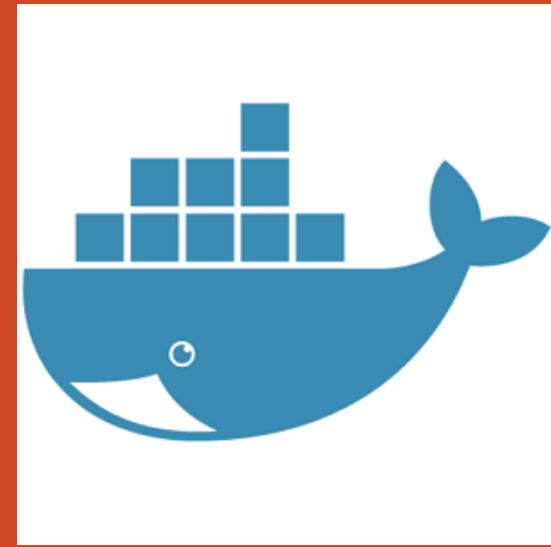
Raw data HTML form

Sigla

Nombre

Creditos

POST



Ajustes para paso a producción y uso de docker-compose

Creamos un generador de SECRET_KEY y modificamos setup.sh y start.sh

```
secret.py x
1 import random
2 secret="".join([random.SystemRandom().choice('abcdefghijklmnopqrstuvwxyz0123456789!@#$%^&*(_+=)') for i in range(50)])
3 print("export SECRET_KEY='{0}'".format(secret))
```

```
setup.sh x
1 #!/bin/bash
2 pip install -r requirements.txt
3 apt-get update
4 apt-get -y install apache2
5 a2enmod proxy
6 a2enmod proxy_http
7 a2enmod ssl
8 rm /etc/apache2/sites-enabled/*
9 cp servicio.conf /etc/apache2/sites-enabled
10 python secret.py >> ~/.bashrc
```

```
start.sh x
1 #!/bin/bash
2 source ~/.bashrc
3 python manage.py makemigrations
4 python manage.py migrate
5 python manage.py collectstatic
6 service apache2 restart
7 gunicorn servicioplanificacion.wsgi
8
```

Editamos production.py para incorporar seguridad

```
production.py x
1  from .settings import *
2
3  DATABASES = {
4      'default': {
5          'ENGINE': 'django.db.backends.mysql',
6          'NAME' : 'planificacion',
7          'USER' : 'icf',
8          'PASSWORD' : 'Secret.123',
9          'DEFAULT-CHARACTER-SET' : 'utf8',
10         'HOST' : '192.168.0.2',
11         'PORT' : '3306',
12         'TEST': {
13             'NAME': 'planificacion_test',
14         }
15     },
16 }
17
18 STATIC_ROOT="/servicioplanificacion/static"
19
20 DEBUG=False
21
22 SECRET_KEY=os.environ["SECRET_KEY"]
23
24 ALLOWED_HOSTS = ['*']
25 USE_X_FORWARDED_HOST=True
26 USE_X_FORWARDED_PORT=True
27
```

Creamos docker-compose.yml

```
docker-compose.yml x
1  version: '3'
2
3  services:
4    server:
5      image: servicio
6      links:
7        - dbserver
8      ports:
9        - "80:80"
10
11   dbserver:
12     image: testdb
13
14   networks:
15     default:
16       external:
17         name: br0
```

Iniciamos, testeamos y detenemos los servicios

```
D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>docker-compose up -d
Creating servicioplanificacion_dbserver_1 ... done
Creating servicioplanificacion_server_1    ... done
```

Create

[OPTIONS](#)[GET](#)

Vista que representa el comportamiento de la API REST.

GET /curso/

```
HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept
```

```
[ ]
```

```
D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>docker-compose stop
Stopping servicioplanificacion_server_1    ... done
Stopping servicioplanificacion_dbserver_1 ... done
```



Amazon ECS/ECR

Creamos un generador de SECRET_KEY y modificamos setup.sh y start.sh para crear un usuario admin

```
secret.py ×
secret.py > ...
1 import random
2 secret="".join([random.SystemRandom().choice('abcdefghijklmnopqrstuvwxyz0123456789!@#$%^&*(_+=)') for i in range(50)])
3 print("export SECRET_KEY='{0}'".format(secret))
4 print("export DJANGO_SUPERUSER_PASSWORD='{0}'".format("Secret.123"))
5 print("export DJANGO_SUPERUSER_EMAIL='{0}'".format("servidor@icfunab.cl"))
6 print("export DJANGO_SUPERUSER_USERNAME='{0}'".format("servidor"))
```

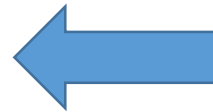
```
start.sh
1 #!/bin/bash
2 source ~/.bashrc
3 python manage.py makemigrations
4 python manage.py migrate
5 python manage.py collectstatic
6 python manage.py createsuperuser --noinput --username $DJANGO_SUPERUSER_USERNAME --email $DJANGO_SUPERUSER_EMAIL
7 service apache2 restart
8 gunicorn servicioplanificacion.wsgi
```

```
setup.sh
1 #!/bin/bash
2 pip install -r requirements.txt
3 apt-get update
4 apt-get -y install apache2
5 a2enmod proxy
6 a2enmod proxy_http
7 a2enmod ssl
8 rm /etc/apache2/sites-enabled/*
9 cp servicio.conf /etc/apache2/sites-enabled
10 python secret.py >> ~/.bashrc
```


Editamos production.py para incorporar seguridad y deshabilitar la conexión a la interfaz de prueba

```
production.py X
servicioplanificacion > production.py > ...
1  from .settings import *
2
3  DATABASES = {
4      'default': {
5          'ENGINE': 'django.db.backends.mysql',
6          'NAME': 'planificacion',
7          'USER': 'icf',
8          'PASSWORD': 'Secret.123',
9          'DEFAULT-CHARACTER-SET': 'utf8',
10         'HOST': 'grupo00.c5d4mi2dthpc.us-east-1.rds.amazonaws.com',
11         'PORT': '3306',
12         'TEST': {
13             'NAME': 'planificacion_test',
14         }
15     },
16 }
17
18 REST_FRAMEWORK = {
19     'DEFAULT_RENDERER_CLASSES': (
20         'rest_framework.renderers.JSONRenderer',
21     )
22 }
```

```
23  STATIC_ROOT="/servicioplanificacion/static"
24  #ADMIN_ENABLED=False
25  DEBUG=False
26  SECRET_KEY=os.environ["SECRET_KEY"]
27
28  ALLOWED_HOSTS = ['*']
29  USE_X_FORWARDED_HOST=True
30  USE_X_FORWARDED_PORT=True
31
32  INSTALLED_APPS += ('corsheaders',)
33  MIDDLEWARE += ('corsheaders.middleware.CorsMiddleware',)
34  CORS_ORIGIN_ALLOW_ALL = True
35
```



Aquí usamos la base de datos en el servidor en amazon RDS

Construimos la imagen y la ejecutamos localmente

Revisamos los logs de inicio con el ID del contenedor

```
build.bat x
build.bat
1 docker build . -t "grupo00"
```

```
run.bat x
run.bat
1 docker run -d -p 80:80 grupo00
```

```
D:\Clases\UNAB\202020\ICF233\servicioplanificacion>docker logs --tail 20 6b81d72c8f42
No changes detected
System check identified some issues:

WARNINGS:
?: (mysql.W002) MySQL Strict Mode is not set for database connection 'default'
    HINT: MySQL's Strict Mode fixes many data integrity problems in MySQL, such as data truncation upon insertion, by escalating warnings into errors. It is strongly recommended you activate it. See: https://docs.djangoproject.com/en/3.1/ref/databases/#mysql-sql-mode
Operations to perform:
  Apply all migrations: admin, api, auth, authtoken, contenttypes, sessions
Running migrations:
  No migrations to apply.

165 static files copied to '/servicioplanificacion/static'.
CommandError: Error: That username is already taken.
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2. Set the 'ServerName' directive globally to suppress this message
Restarting Apache httpd web server: apache2.
[2020-11-25 17:01:07 +0000] [95] [INFO] Starting gunicorn 20.0.4
[2020-11-25 17:01:07 +0000] [95] [INFO] Listening at: http://127.0.0.1:8000 (95)
[2020-11-25 17:01:07 +0000] [95] [INFO] Using worker: sync
[2020-11-25 17:01:07 +0000] [98] [INFO] Booting worker with pid: 98
```

La migración y la creación del usuario solamente ocurren la primera vez que se sube el contenedor o si borramos la base de datos

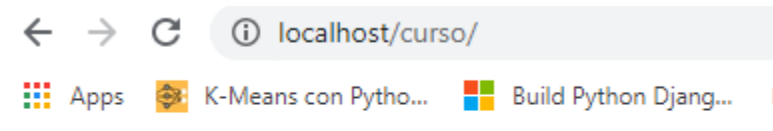
Testeamos el servicio en el contenedor local usando curl

```
curl -X POST -F "username=servidor" -F "password=Secret.123" http://localhost/api/login
```

```
D:\Clases\UNAB\202020\ICF233\servicioplanificacion>curl -X POST -F "username=servidor" -F "password=Secret.123" http://localhost/api/login  
{ "token": "bf53bc1a0a6ce0fd900b370f84b7f114eef1e7bb" }
```

```
curl -X POST -F "sigla=ICF233" -F "nombre=IngenieriaSoftwareII" -F "creditos=4" -H "token: bf53bc1a0a6ce0fd900b370f84b7f114eef1e7bb" http://localhost/curso/
```

```
D:\Clases\UNAB\202020\ICF233\servicioplanificacion>curl -X POST -F "sigla=ICF233" -F "nombre=IngenieriaSoftwareII" -F "creditos=4" -H "token: bf53bc1a0a6ce0fd900b370f84b7f114eef1e7bb" http://localhost/curso/  
{ "id": 2, "sigla": "ICF233", "nombre": "IngenieriaSoftwareII", "creditos": 4 }
```



```
curl -X GET -H "token: bf53bc1a0a6ce0fd900b370f84b7f114eef1e7bb" http://localhost/curso/
```

Instalar aws cli y etiquetar la imagen

<https://docs.aws.amazon.com/cli/latest/userguide/install-cliv2-windows.html>

```
D:\Clases\UNAB\202020\ICF233\servicioplanificacion>docker image ls
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
grupo00	latest	3d63f5b37935	14 minutes ago	955MB
python	3.7	94c9a318e47a	7 days ago	876MB
optic	latest	c2918afc5a2e	5 weeks ago	516MB
nvidia/cuda	10.2-base	038eb67e1704	7 weeks ago	107MB
ubuntu	bionic	56def654ec22	2 months ago	63.2MB

```
D:\Clases\UNAB\202020\ICF233\servicioplanificacion>docker tag 3d63f5b37935 91380866659.dkr.ecr.us-east-1.amazonaws.com/grupo00
```

```
D:\Clases\UNAB\202020\ICF233\servicioplanificacion>aws configure
AWS Access Key ID [None]: AKIA5JQZ5WARUYSK4VNI
AWS Secret Access Key [None]: 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
Default region name [us-east-1]: us-east-1
Default output format [None]:
```

Autenticar docker con Amazon ECR y hacer el push

```
aws ecr get-login-password | docker login --username AWS --password-stdin  
91380866659.dkr.ecr.us-east-1.amazonaws.com/grupo00
```

```
docker push 91380866659.dkr.ecr.us-east-1.amazonaws.com/grupo00
```

```
D:\Clases\UNAB\202020\ICF233\servicioplanificacion>aws ecr get-login-password | docker login --username AWS --password-stdin 91380866659.dkr.ecr.  
us-east-1.amazonaws.com/grupo00  
Login Succeeded
```













```
D:\Clases\UNAB\202020\ICF233\servicioplanificacion>docker push 91380866659.dkr.ecr.us-east-1.amazonaws.com/grupo00  
The push refers to repository [91380866659.dkr.ecr.us-east-1.amazonaws.com/grupo00]
```

```
7ea290ac2c93: Pushed  
b4fcf22d25d0: Pushed  
53994b9a2da5: Pushed  
adec77bceebb: Pushed  
026c477245c5: Pushed  
ee78bcfefc78: Pushed  
c4a6d8ca5d2c: Pushed  
059ed1793a98: Pushed  
712264374d24: Pushed  
475b4eb79695: Pushed  
f3be340a54b9: Pushed  
114ca5b7280f: Pushed  
latest: digest: sha256:b1862699acdf3ddc7ba531162ada5c5a8ca13180240888eb00eca6eea7b76d06 size: 2846
```

Copiar el contenedor a otro computador como archivo

Usamos *docker save* para guardar la imagen

```
D:\Clases\UNAB\202020\ICF233\servicioplanificacion>docker save grupo00 -o grupo00.docker
```

 grupo00.docker	30/11/2020 15:35	DOCKER File	961,843 KB
 secret.py	25/11/2020 14:14	Python File	1 KB
 run.bat	25/11/2020 14:00	Windows Batch File	1 KB
 start.sh	25/11/2020 13:50	Shell Script	1 KB
 build.bat	25/11/2020 12:43	Windows Batch File	1 KB
 README.MD	13/10/2020 18:19	MD File	1 KB
 .coverage	13/10/2020 17:24	Code Coverage X...	3 KB
 requirements.txt	13/6/2019 11:12	Text Document	1 KB
 Procfile	11/6/2019 21:26	File	1 KB
 heroku_secret.py	11/6/2019 20:59	Python File	1 KB
 docker-compose.yml	11/6/2019 20:13	YML File	1 KB
 setup.sh	11/6/2019 19:28	Shell Script	1 KB

Copiar el contenedor a otro computador como archivo

Usamos *docker load* para recuperar la imagen

```
D:\Clases\UNAB\202020\ICF233\servicioplanificacion>docker load -i grupo00.docker
Loaded image: grupo00:latest
```

```
D:\Clases\UNAB\202020\ICF233\servicioplanificacion>docker image ls
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
ubuntu	latest	f643c72bc252	4 days ago	72.9MB
91380866659.dkr.ecr.us-east-1.amazonaws.com/grupo00	latest	3d63f5b37935	5 days ago	955MB
grupo00	latest	3d63f5b37935	5 days ago	955MB
python	3.7	94c9a318e47a	12 days ago	876MB

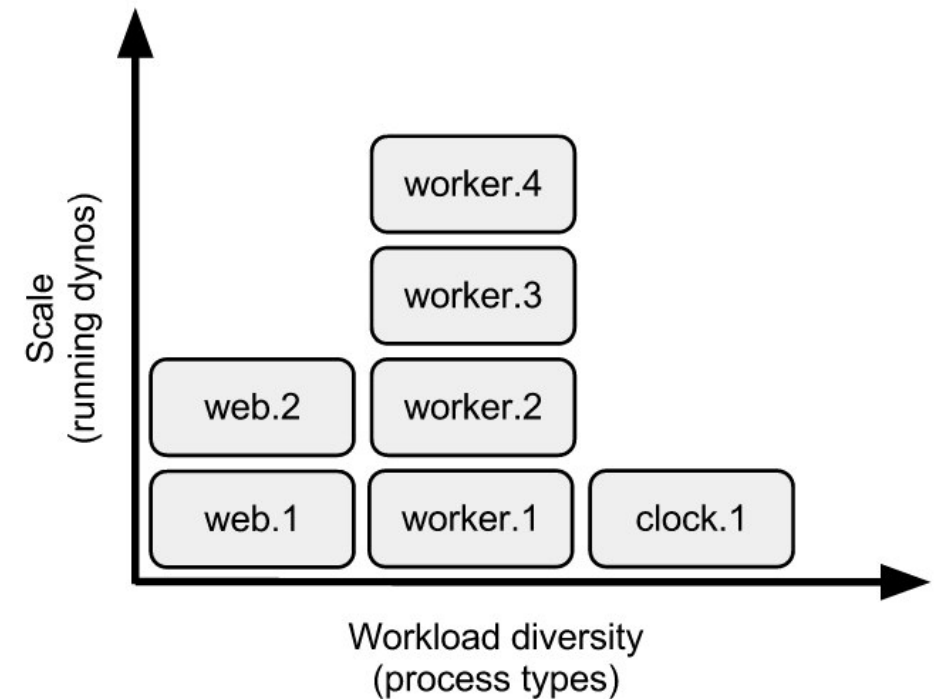
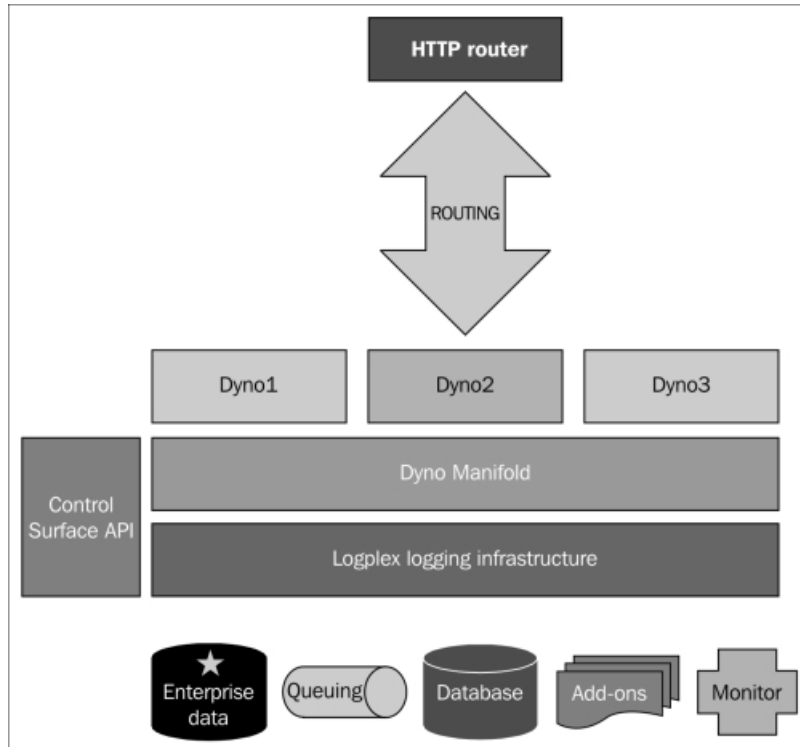
```
D:\Clases\UNAB\202020\ICF233\servicioplanificacion>docker run -d -p 80:80 grupo00
217737f14b39b7827cb9ab85862237b0bcf46832872820b699e5977c7903f4b6
```

```
D:\Clases\UNAB\202020\ICF233\servicioplanificacion>curl -X POST -F "username=servidor" -F "password=Secret.123" http://localhost/api/login
{"token":"bf53bc1a0a6ce0fd900b370f84b7f114eef1e7bb"}
```



Heroku

¿Qué es heroku?



<https://www.heroku.com/>
<https://devcenter.heroku.com/articles/getting-started-with-python>
<https://devcenter.heroku.com/articles/how-heroku-works>
<https://devcenter.heroku.com/articles/dyno-types>
<https://hub.docker.com/r/finalgene/heroku-cli/>



Creamos el Procfile, agregamos django-heroku a requirements.txt y lo instalamos

```
Procfile x
1 web: gunicorn servicioplanificacion.wsgi
```

La P mayúscula inicial es importante en el nombre de Procfile

```
requirements.txt x
1 django
2 djangorestframework
3 gunicorn
4 mysqlclient
5 coverage
6 django-heroku
```

```
(venv) D:\Clases\UNAB\ICF232-201910\servicios>pip3 install django-heroku
Collecting django-heroku
```

Creamos una nueva setting: heroku.py

```
heroku.py x
1  from .settings import *
2  import django_heroku
3
4  DATABASES = {
5      'default': {
6          'ENGINE': 'django.db.backends.sqlite3',
7          'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
8      },
9  }
10
11  DEBUG=False
12
13  SECRET_KEY=os.environ["SECRET_KEY"]
14
15  ALLOWED_HOSTS = ['*']
16  USE_X_FORWARDED_HOST=True
17  USE_X_FORWARDED_PORT=True
18
19  PROJECT_ROOT = os.path.dirname(os.path.abspath(__file__))
20  STATIC_URL = '/static/'
21  STATIC_ROOT = os.path.join(PROJECT_ROOT, 'static')
22  django_heroku.settings(locals())
```

Creamos la aplicación en heroku, el archivo heroku_secret.py, lo ejecutamos y configuramos variables de entorno

```
heroku_secret.py x
1 import random
2 secret="".join([random.SystemRandom().choice('abcdefghijklmnopqrstuvwxyz0123456789!@#$%^&*(_+=)') for i in range(50)])
3 print("heroku config:set SECRET_KEY=\"{0}\"".format(secret))
```

```
(venv) D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>heroku create
Creating app... done, ☑ sleepy-gorge-24969
https://sleepy-gorge-24969.herokuapp.com/ | https://git.heroku.com/sleepy-gorge-24969.git
```

```
(venv) D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>python heroku_secret.py
heroku config:set SECRET_KEY="j)%5#d=u$-+pei2c@krq&04&@n**+-q2!y_+l5e@dogk8r_&#+"
```

```
(venv) D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>heroku config:set SECRET_KEY="j)%5#d=u$-+pei2c@krq&04&@n**+-q2!y_+l5e@dogk8r_&#+"
Setting SECRET_KEY and restarting ☑ sleepy-gorge-24969... done, v4
SECRET_KEY: j)%5#d=u$-+pei2c@krq&04&@n**+-q2!y_+l5e@dogk8r_&#+
```

```
(venv) D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>heroku config:set DJANGO_SETTINGS_MODULE=servicioplanificacion.heroku
Setting DJANGO_SETTINGS_MODULE and restarting ☑ sleepy-gorge-24969... done, v5
DJANGO_SETTINGS_MODULE: servicioplanificacion.heroku
```

Hacemos commit de los cambios y enviamos la aplicación

```
(venv) D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>git add .
```

```
(venv) D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>git commit -m "heroku"
[master a50fc25] heroku
1 file changed, 1 insertion(+)
```

```
(venv) D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>git push heroku master
Counting objects: 4, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 430 bytes | 215.00 KiB/s, done.
Total 4 (delta 2), reused 0 (delta 0)
remote: Compressing source files... done.
remote: Building source:
remote:
remote: -----> Python app detected
remote: -----> Installing requirements with pip
remote:
remote: -----> $ python manage.py collectstatic --noinput
remote:          152 static files copied to '/tmp/build_4ce57f24b96248bde84530d36ad6b73c/staticfiles', 454 post-processed.
remote:
remote: -----> Discovering process types
remote:          Procfile declares types -> web
remote:
remote: -----> Compressing...
remote:          Done: 55.7M
remote: -----> Launching...
remote:          Released v12
remote:          https://sleepy-gorge-24969.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy... done.
To https://git.heroku.com/sleepy-gorge-24969.git
   86d8a81..5dadd42 master -> master
```

Preparamos la aplicación

```
(venv) D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>heroku run python manage.py makemigrations
Running python manage.py makemigrations on ☐ sleepy-gorge-24969... up, run.8247 (Free)
No changes detected
```

```
(venv) D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>heroku run python manage.py migrate
Running python manage.py migrate on ☐ sleepy-gorge-24969... up, run.2041 (Free)
```

Operations to perform:

Apply all migrations: admin, api, auth, authtoken, contenttypes, sessions

Running migrations:

```
Applying contenttypes.0001_initial... OK
Applying auth.0001_initial... OK
Applying admin.0001_initial... OK
Applying admin.0002_logentry_remove_auto_add... OK
Applying admin.0003_logentry_add_action_flag_choices... OK
Applying api.0001_initial... OK
Applying api.0002_auto_20190507_2306... OK
Applying contenttypes.0002_remove_content_type_name... OK
Applying auth.0002_alter_permission_name_max_length... OK
Applying auth.0003_alter_user_email_max_length... OK
Applying auth.0004_alter_user_username_opts... OK
Applying auth.0005_alter_user_last_login_null... OK
Applying auth.0006_require_contenttypes_0002... OK
Applying auth.0007_alter_validators_add_error_messages... OK
Applying auth.0008_alter_user_username_max_length... OK
Applying auth.0009_alter_user_last_name_max_length... OK
Applying auth.0010_alter_group_name_max_length... OK
Applying auth.0011_update_proxy_permissions... OK
Applying authtoken.0001_initial... OK
Applying authtoken.0002_auto_20160226_1747... OK
Applying sessions.0001_initial... OK
```

Definimos la cantidad de procesos, creamos el superusuario y probamos la api con curl

```
(venv) D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>heroku ps:scale web=1
Scaling dynos... done, now running web at 1:Free
```

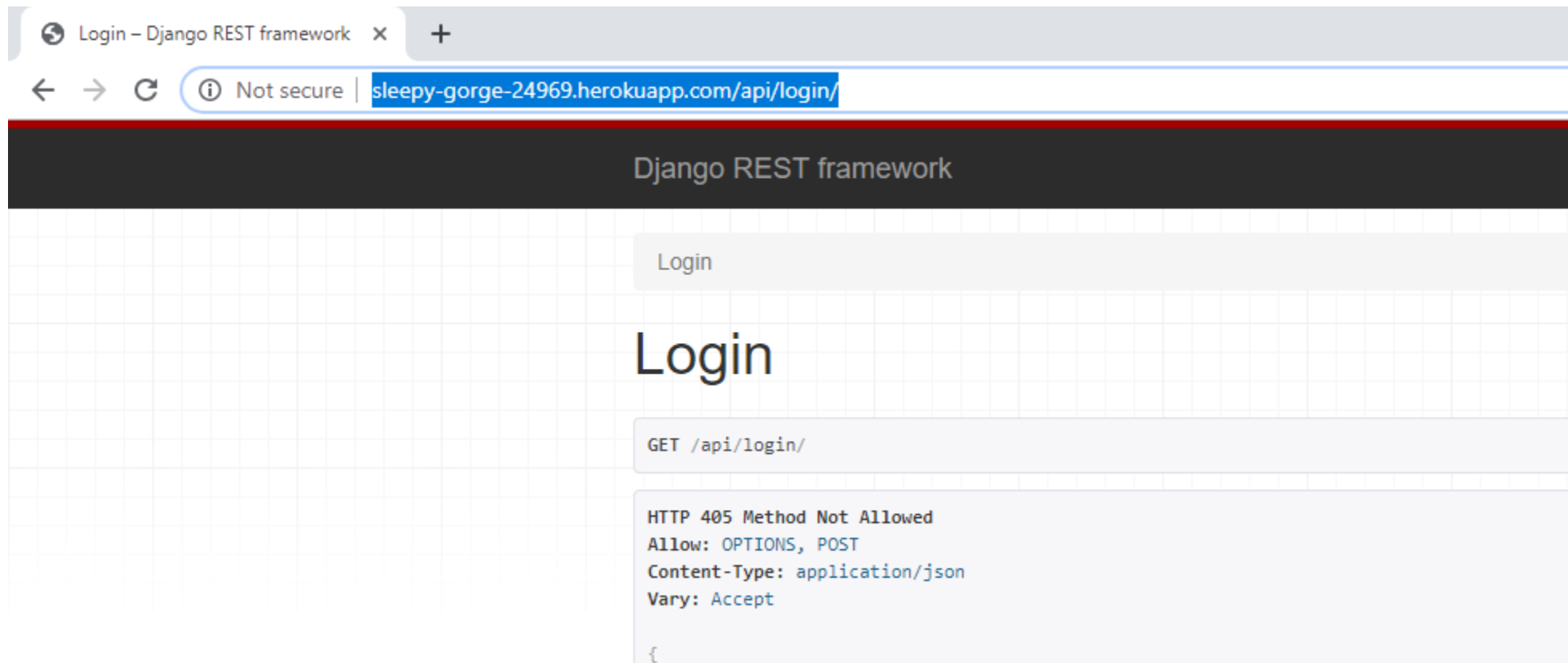
```
(venv) D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>heroku run python manage.py createsuperuser
Running python manage.py createsuperuser on ☐ sleepy-gorge-24969... up, run.7799 (Free)
Username (leave blank to use 'u45763'): servidor
Email address:
Password:
Password (again):
Superuser created successfully.
```

```
curl -X POST -F "username=servidor" -F "password=Secret.123" http://sleepy-gorge-24969.herokuapp.com/api/login
```

```
(venv) D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>curl -X POST -F "username=servidor" -F "password=Secret.123" http://sleepy-gorge-24969.herokuapp.com/api/login
{"token":"327891e38c89a354bafc4df85ff9c93a5b771bd5"}
```

Si obtenemos un token nuestra aplicación está funcionando

También podemos revisarla en un browser en el dominio herokuapp.com



Para permitir acceso desde otras aplicaciones necesitamos habilitar CORS en nuestro servicio

```
requirements.txt x
requirements.txt
1  django
2  djangorestframework
3  gunicorn
4  mysqlclient
5  coverage
6  django-heroku
7  django-cors-headers
```

Agregamos django-cors-headers a requirements.txt

Incorporamos la configuración en production.py y heroku.py

production.py x

servicioplanificacion > production.py > ...

```
1  from .settings import *
2
3  DATABASES = {
4      'default': {
5          'ENGINE': 'django.db.backends.mysql',
6          'NAME' : 'planificacion',
7          'USER' : 'icf',
8          'PASSWORD' : 'Secret.123',
9          'DEFAULT-CHARACTER-SET' : 'utf8',
10         'HOST' : '192.168.0.2',
11         'PORT' : '3306',
12         'TEST': {
13             'NAME': 'planificacion_test',
14         }
15     },
16 }
17
18 STATIC_ROOT="/servicioplanificacion/static"
19
20 DEBUG=False
21
22 SECRET_KEY=os.environ["SECRET_KEY"]
23
24 ALLOWED_HOSTS = ['*']
25 USE_X_FORWARDED_HOST=True
26 USE_X_FORWARDED_PORT=True
27
28 INSTALLED_APPS += ('corsheaders',)
29 MIDDLEWARE += ('corsheaders.middleware.CorsMiddleware',)
30 CORS_ORIGIN_ALLOW_ALL = True
```

heroku.py x

servicioplanificacion > heroku.py > ...

```
1  from .settings import *
2  import django_heroku
3
4  DATABASES = {
5      'default': {
6          'ENGINE': 'django.db.backends.sqlite3',
7          'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
8      },
9  }
10
11 DEBUG=False
12
13 SECRET_KEY=os.environ["SECRET_KEY"]
14
15 ALLOWED_HOSTS = ['*']
16 USE_X_FORWARDED_HOST=True
17 USE_X_FORWARDED_PORT=True
18
19 PROJECT_ROOT = os.path.dirname(os.path.abspath(__file__))
20 STATIC_URL = '/static/'
21 STATIC_ROOT = os.path.join(PROJECT_ROOT, 'static')
22 django_heroku.settings(locals())
23
24 INSTALLED_APPS += ('corsheaders',)
25 MIDDLEWARE += ('corsheaders.middleware.CorsMiddleware',)
26 CORS_ORIGIN_ALLOW_ALL = True
```

Hacemos commit y recompilamos la aplicación

```
(venv) D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>git add .

(venv) D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>git commit -m "cors"
[master 174fa14] cors
 3 files changed, 10 insertions(+), 1 deletion(-)

(venv) D:\Clases\UNAB\ICF232-201910\servicios\servicioplanificacion>git push heroku master
Counting objects: 6, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 696 bytes | 174.00 KiB/s, done.
Total 6 (delta 5), reused 0 (delta 0)
remote: Compressing source files... done.
remote: Building source:
remote:
remote: -----> Python app detected
```

Preguntas

