

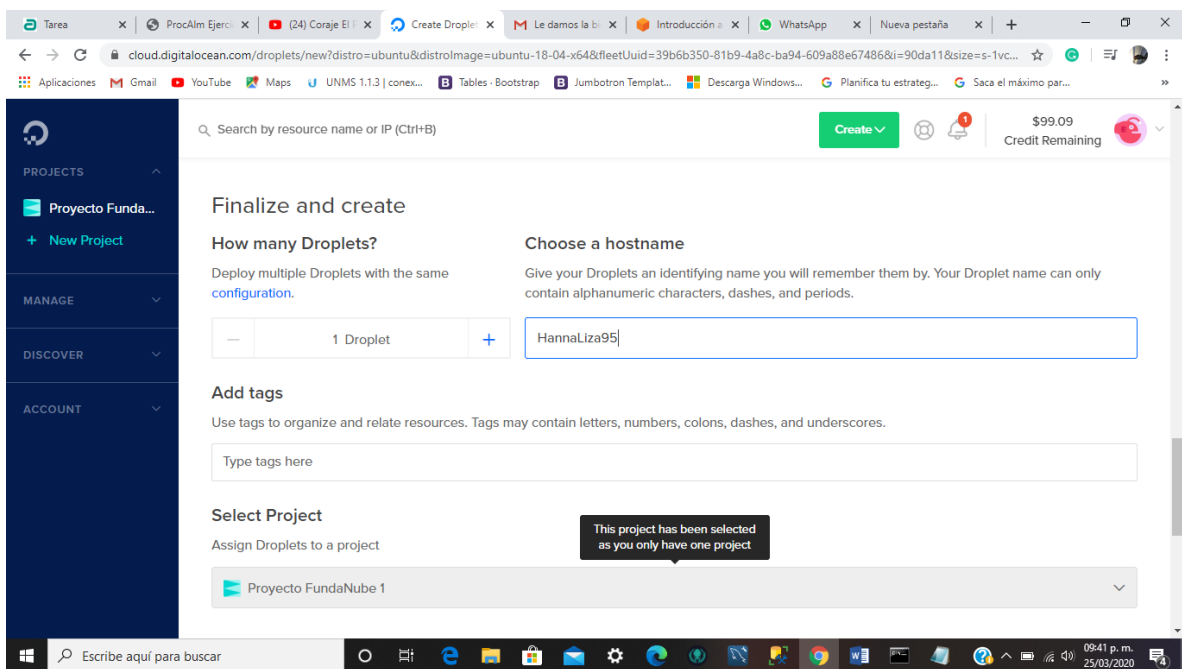
## 1. Crear Droplet, contemplando todos los parámetros básicos

The screenshot shows the DigitalOcean 'Create Droplet' interface. On the left is a sidebar with navigation links: PROJECTS, MANAGE, DISCOVER, and ACCOUNT. The main content area is titled 'Create Droplet' and includes a search bar. Below the search bar, there are tabs for different operating systems: Ubuntu (selected), FreeBSD, Fedora, Debian, and CentOS. Each tab has a 'Select version' dropdown. The 'Ubuntu' tab shows version '18.04.3 (LTS) x64'. Below the OS tabs, there's a 'Choose a plan' section with two main categories: 'STARTER' and 'PERFORMANCE'. Under 'STARTER', the 'Standard' plan is selected, showing a price of \$5/mo. Under 'PERFORMANCE', there are three options: 'General Purpose', 'CPU-Optimized', and 'Memory-Optimized' (marked as 'NEW'). Below the plans, there's a description of the 'Standard' plan: 'Standard virtual machines with a mix of memory and compute resources. Best for small projects that can handle variable levels of CPU performance, like blogs, web apps and dev/test environments.' At the bottom, there are six pricing cards for different configurations, ranging from \$5/mo to \$20/mo.

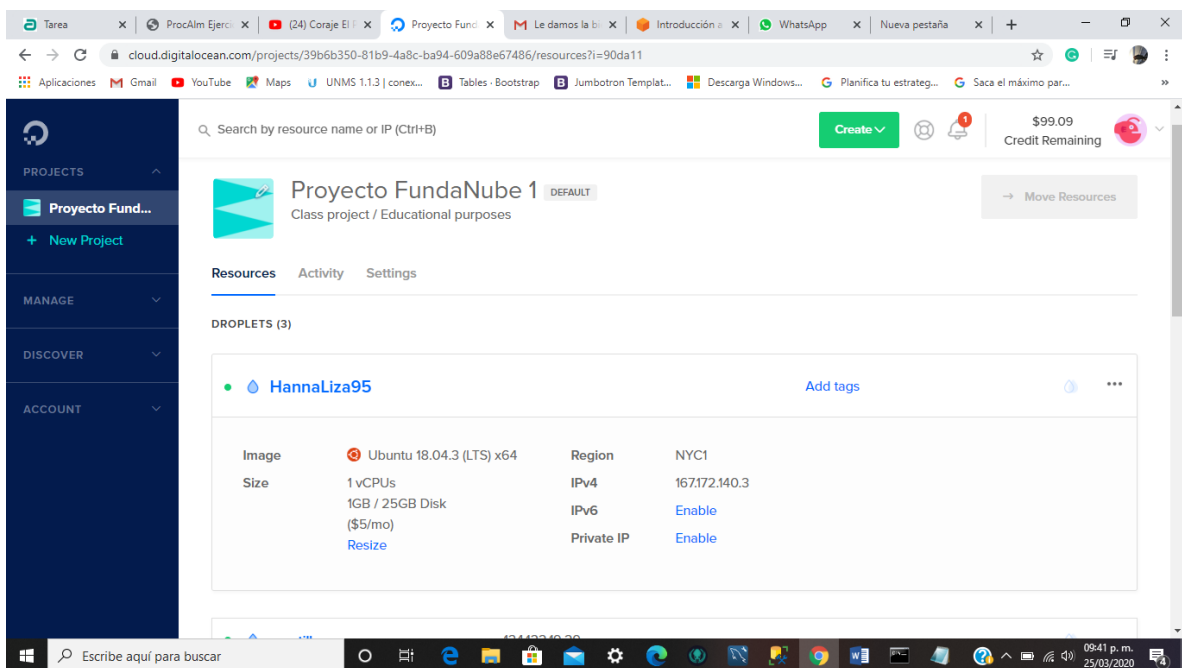
## 2. Generar una ssh key

The screenshot shows the DigitalOcean 'Authentication' page. The 'SSH keys' section is active, showing a 'New SSH Key' button. Below the button, there's a section titled 'Finalize and create' with a question 'How many Droplets?' and a dropdown menu set to '1 Droplet'. A text area displays the generated SSH key. A note window is open over the key, showing the key content: 'ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCY1yBY7Ss8FY6Y+cLQfgn++8KBC/iUnvReQ+/8saVdhIpdhzbzdyG1V'. The note window has a title bar 'id\_rsa: Bloc de notes' and a menu bar with 'Archivo', 'Edición', 'Formato', and 'Ver Ayuda'.

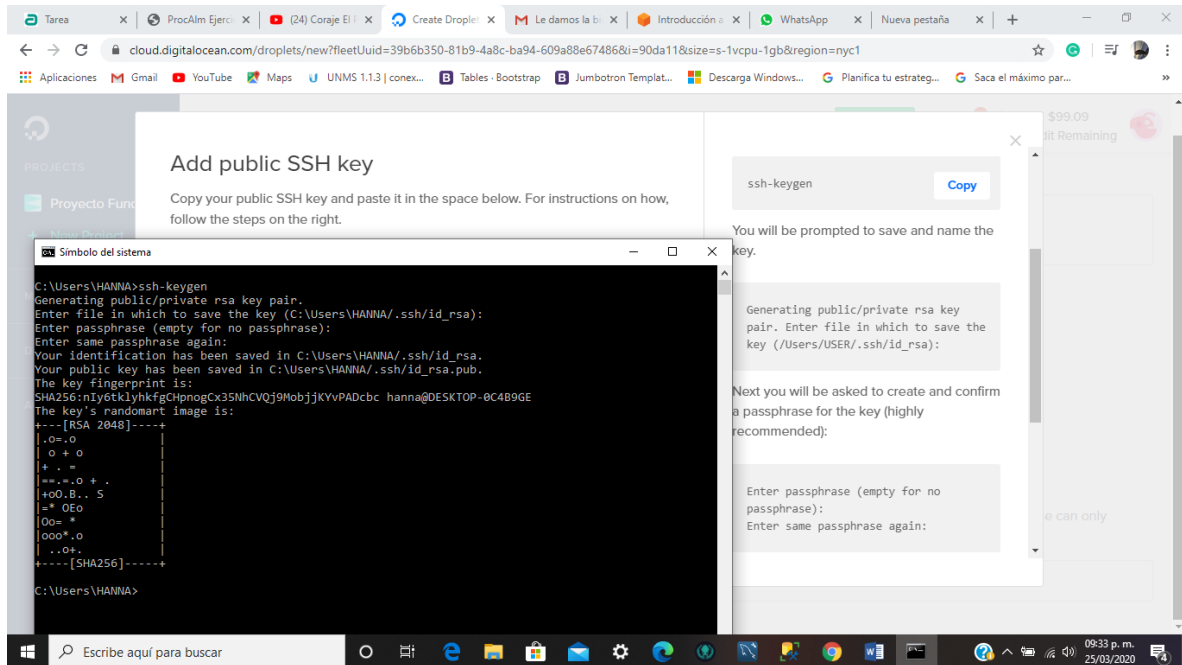
### 3. Ponerle un nombre al droplet



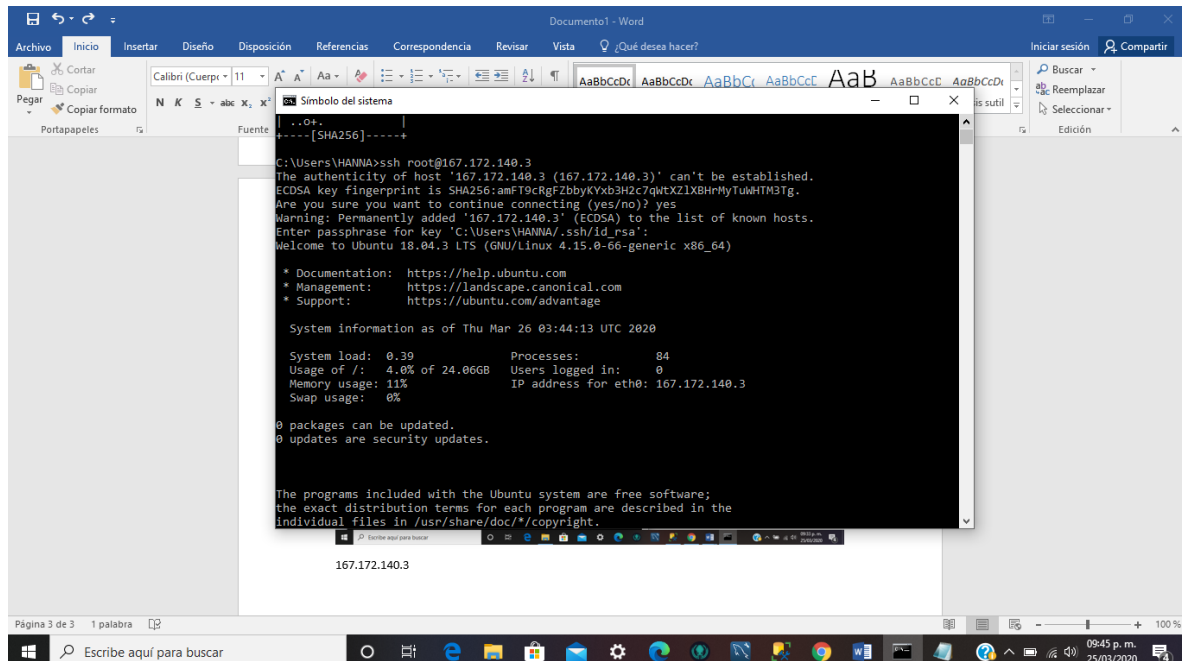
### 4. Ya que le damos crear el droplet, nos debe aparecer así, esa ipv4 es la que vamos a requerir durante todo el proceso



## 5. Primero entramos a nuestra consola y utilizamos la ssh key



## 6. Luego ingresamos utilizando ssh [root@167.172.140.3](https://root@167.172.140.3) la cual es mi IP que me dio digital ocean, ingresamos nuestra contraseña y avanzamos.



7. Al ingresar al servidor nos debe ya aparecer con el nombre de nuestro dropet y ya podemos empezar a utilizar linux

```
root@HannaLiza95: ~  
* Canonical Livepatch is available for installation.  
- Reduce system reboots and improve kernel security. Activate at:  
https://ubuntu.com/livepatch  
119 packages can be updated.  
73 updates are security updates.  
Last login: Fri Mar 27 02:09:31 2020 from 187.190.168.252  
root@HannaLiza95:~#
```

```
root@HannaLiza95: ~  
Microsoft Windows [Versión 10.0.18363.720]  
(c) 2019 Microsoft Corporation. Todos los derechos reservados.  
C:\Users\HANNA>ssh root@167.172.140.3  
Enter passphrase for key 'C:\Users\HANNA\.ssh\id_rsa':  
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-66-generic x86_64)  
  
* Documentation: https://help.ubuntu.com  
* Management: https://landscape.canonical.com  
* Support: https://ubuntu.com/advantage  
  
System information as of Fri Mar 27 03:00:16 UTC 2020  
  
System load: 0.0 Processes: 85  
Usage of /: 4.9% of 24.06GB Users logged in: 1  
Memory usage: 15% IP address for eth0: 167.172.140.3  
Swap usage: 0%  
  
* Canonical Livepatch is available for installation.  
- Reduce system reboots and improve kernel security. Activate at:  
https://ubuntu.com/livepatch  
119 packages can be updated.  
73 updates are security updates.  
Last login: Fri Mar 27 02:09:31 2020 from 187.190.168.252  
root@HannaLiza95:~# sudo apt-get remove docker docker-engine.io containerd runc  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
E: Unable to locate package docker-engine.io  
E: Couldn't find any package by glob 'docker-engine.io'  
E: Couldn't find any package by regex 'docker-engine.io'  
root@HannaLiza95:~# sudo apt-get update  
Get:1 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]  
Get:2 http://mirrors.digitalocean.com/ubuntu bionic InRelease [242 kB]  
Hit:3 http://mirrors.digitalocean.com/ubuntu bionic-updates InReleaseHit:4 http://mirrors.digitalocean.com/ubuntu bionic-backports InRelease  
Fetched 331 kB in 1s (492 kB/s)  
Reading package lists... Done  
root@HannaLiza95:~#
```

8. Para instalar Docker, primero desinstalar cualquier versión anterior, si es que la hay.

```
root@HannaLiza95:~# sudo apt-get remove docker docker-engine.io containerd runc
```

9. Actualizar el paquete apt

```
root@HannaLiza95:~# sudo apt-get update  
Get:1 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]  
Get:2 http://mirrors.digitalocean.com/ubuntu bionic InRelease [242 kB]  
Hit:3 http://mirrors.digitalocean.com/ubuntu bionic-updates InReleaseHit:4 http://mirrors.digitalocean.com/ubuntu bionic-backports InRelease  
Fetched 331 kB in 1s (492 kB/s)  
Reading package lists... Done
```

10. Instalar paquetes para permitir que apt pueda utilizar un repositorio mediante HTTPS

```
root@HannaLiza95:~# sudo apt-get install \
> apt-transport-https \
> ca-certificates \
> curl \
> gnupg-agent \
> software-properties-common
```

11. Agregar la llave GPG oficial de Docker

```
root@HannaLiza95:~# curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

12. Configurar un repositorio estable

```
root@HannaLiza95:~# sudo add-apt-repository \
> "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
> $(lsb_release -cs) \
> stable"
```

13. Actualizar el paquete apt

```
root@HannaLiza95:~# sudo apt-get update
```

14. Instalar la última versión de Docker

```
root@HannaLiza95:~# sudo apt-get install docker-ce docker-ce-cli containerd.io
```

15. Correr el comando 'docker ps' para corroborar que se encuentre instalado correctamente el Docker

```
root@HannaLiza95:~# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
--------------	-------	---------	---------	--------	-------	-------

16. Para descargar la última versión de Docker Compose, se corre el siguiente comando

```
root@HannaLiza95:~# sudo curl -L "https://github.com/docker/compose/releases/download/1.25.4/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
```

% Total	% Received	% Xferd	Average Speed	Time	Time	Time	Current
			Dload	Upload	Total	Spent	Left
0	0	0	0	0	0	0	100
100	16.3M	100	16.3M	0	49.3M	0	49.3M

17. Aplicar permisos

```
root@HannaLiza95:~# sudo chmod +x /usr/local/bin/docker-compose
```

18. Verificar versión de Docker Compose para corroborar que se haya instalado correctamente

```
root@HannaLiza95:~# docker-compose --version
docker-compose version 1.25.4, build 8d51620a
```

19. Clonar el repositorio de GitHub con nuestros archivos, puede ser en cualquier carpeta

```
root@HannaLiza95: /home

root@HannaLiza95:/home# mkdir repos_linux
root@HannaLiza95:/home# ls -l
total 4
drwxr-xr-x 2 root root 4096 Mar 27 03:46 repos_linux
root@HannaLiza95:/home#

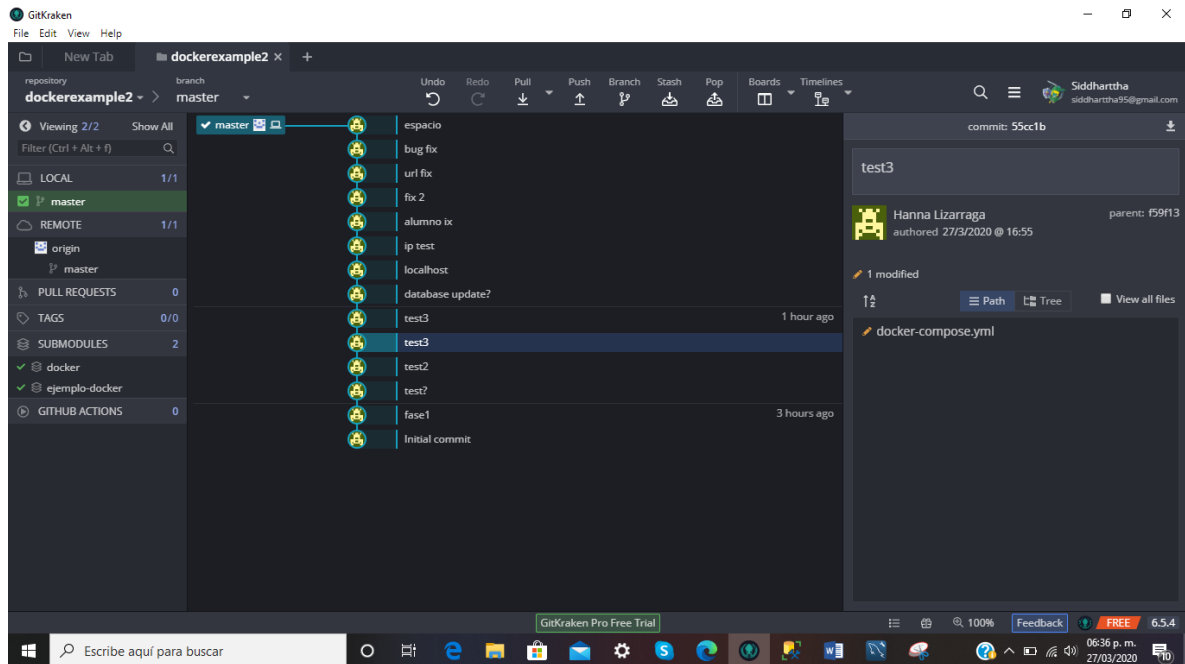
root@HannaLiza95: /home/repos_linux

root@HannaLiza95:/home/repos_linux# git clone https://github.com/HannaSiddhartha/mvc2.git
```

20. Aquí podemos ver que se clono con éxito

```
root@HannaLiza95: /home/repos_linux

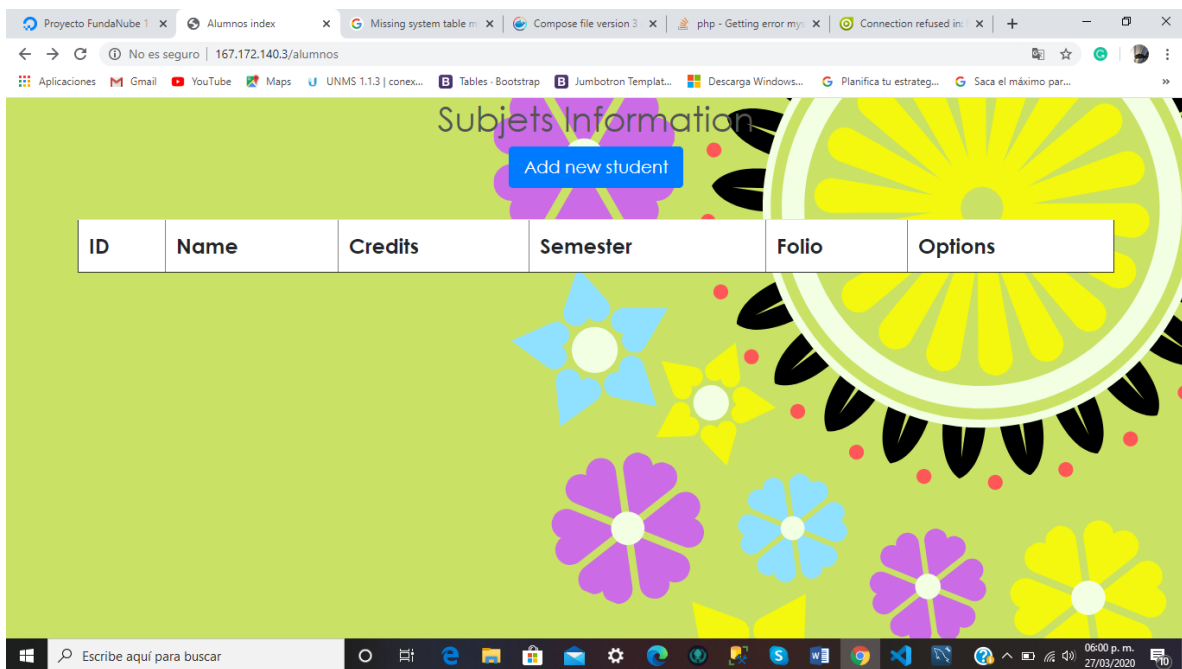
root@HannaLiza95:/home/repos_linux# git clone https://github.com/HannaSiddhartha/mvc2.git
Cloning into 'mvc2'...
remote: Enumerating objects: 530, done.
remote: Counting objects: 100% (530/530), done.
remote: Compressing objects: 100% (244/244), done.
remote: Total 530 (delta 300), reused 506 (delta 279), pack-reused 0
Receiving objects: 100% (530/530), 19.01 MiB | 22.43 MiB/s, done.
Resolving deltas: 100% (300/300), done.
root@HannaLiza95:/home/repos_linux#
```



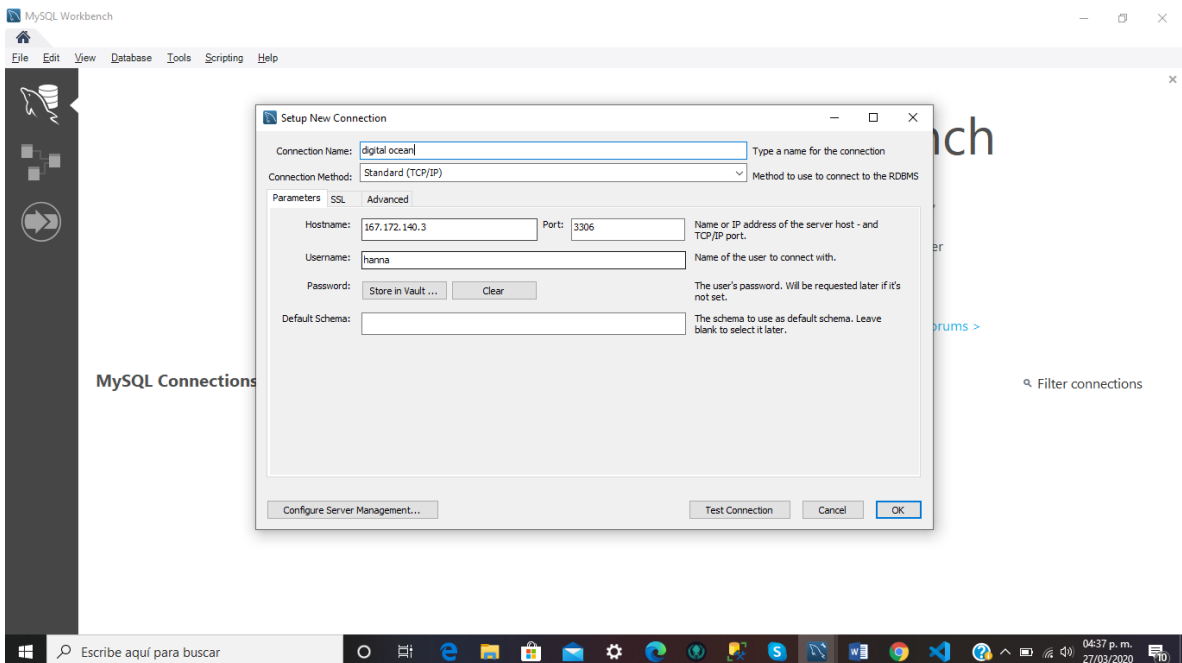
## 21. Levantar el contenedor mediante el docker-compose

```
root@HannaLiza95:/home/repos_linux# docker-compose up
```

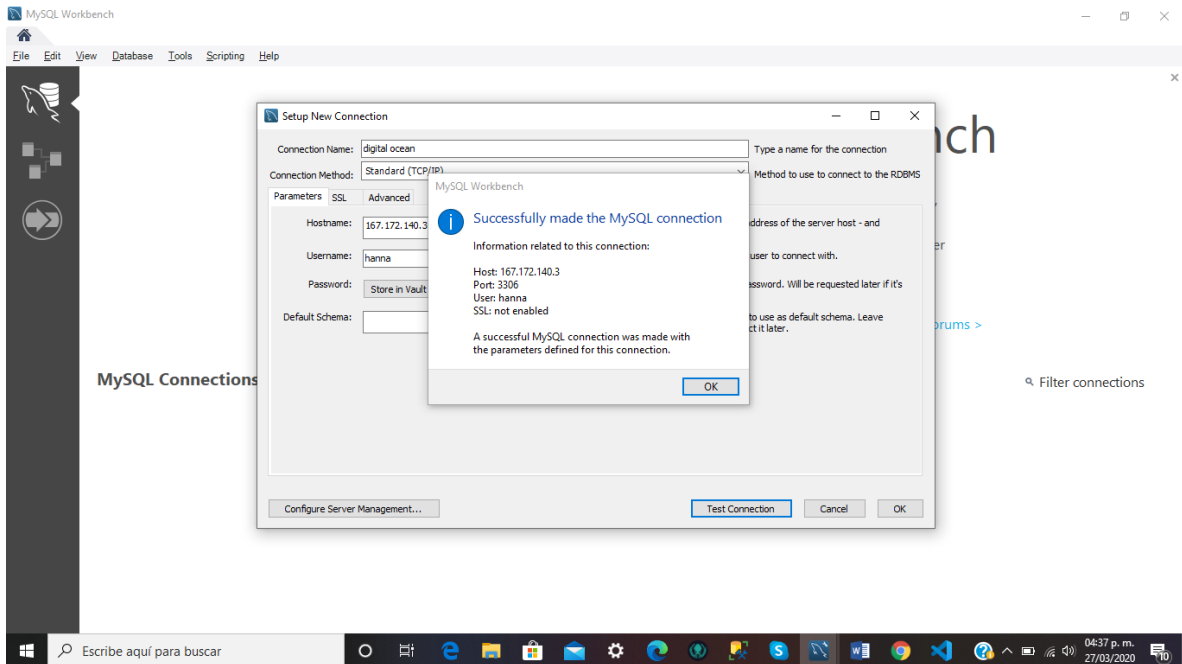
## 22. Después de este punto, ya se podrá acceder al contenedor mediante un explorador de internet



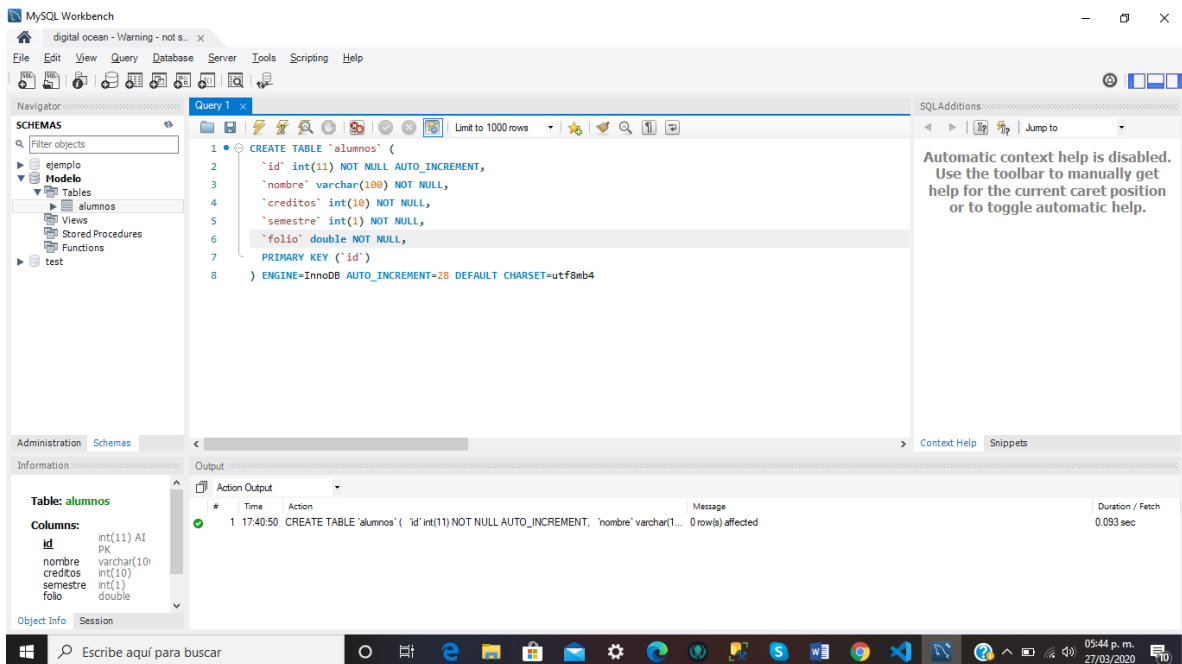
## 23. Se configura la conexión a la base de datos en MySQL Workbench



## 24. Se configuro con éxito



## 25. Ahora, ya se podrá modificar la base de datos.





26. Actualizar el archivo index.html

```
<body>
<center>
<font color="#545454"size="10" face="Century Gothic">
  <h1>Students Information</h1>
</font>
</center>
<center>
<font color="#545454"size="5" face="Century Gothic">
<a href="alumnos/insert"class="btn btn-primary btn-lg">Add new student</a>
```

27. Desde el contenedor, hacer un Pull para actualizar el archivo

```
root@HannaLiza95:~# cd /home/repos_linux/dockerexample
root@HannaLiza95:/home/repos_linux/dockerexample# git pull
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 6 (delta 4), reused 2 (delta 0), pack-reused 0
Unpacking objects: 100% (6/6), done.
From https://github.com/HannaSiddhartha/dockerexample
   a4fbee4..9758f64  master    -> origin/master
Updating a4fbee4..9758f64
Fast-forward
 flerky/application/controllers/Alumnos.php | 2 +-

```

28. Al acceder en el contenedor en el explorador, se pueden apreciar los cambios (Deploy).

