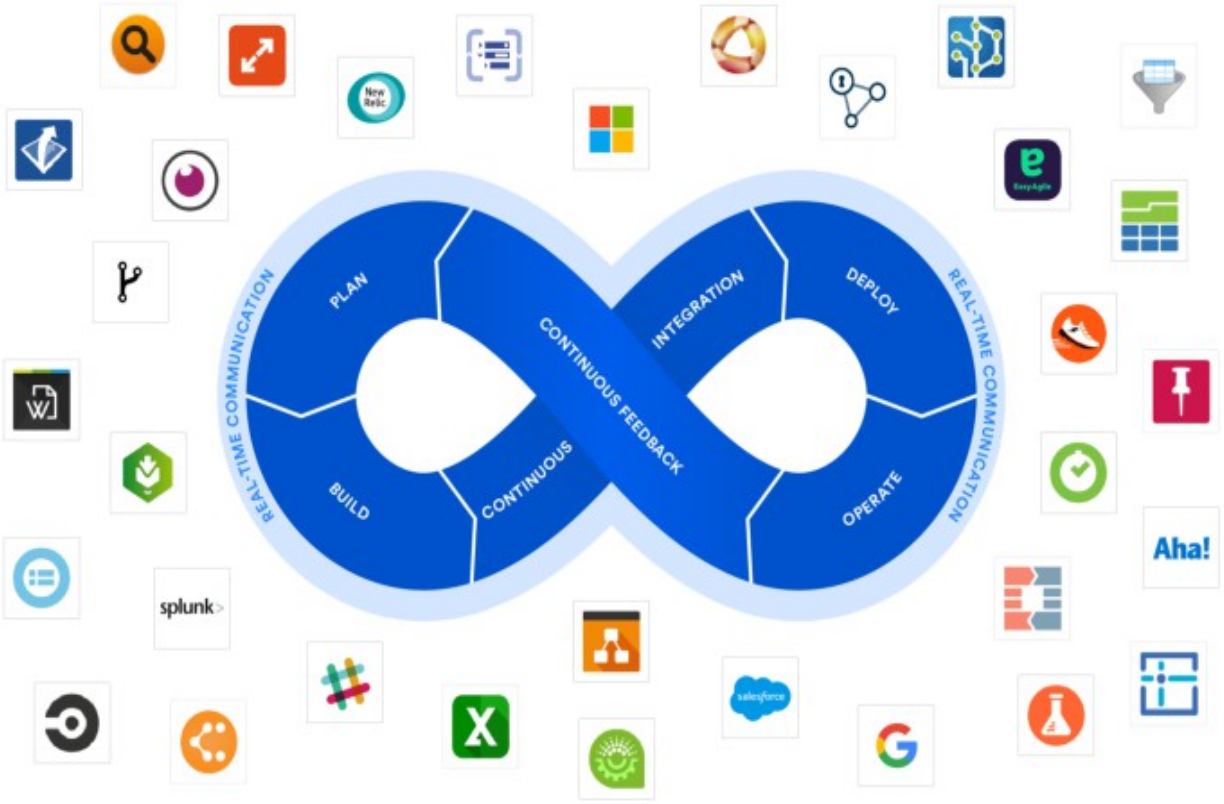
# Universidad Modelo

## Creación de un Droplet




**Ing en Desarrollo de Tecnologías y Software**

**Fundamentos de la Nube**

**Docente:**
**Ing. Alfredo Jose Bolio Domínguez**

**Alumno:**
**Carlos Iván Agüero Pérez**
**28/03/20**

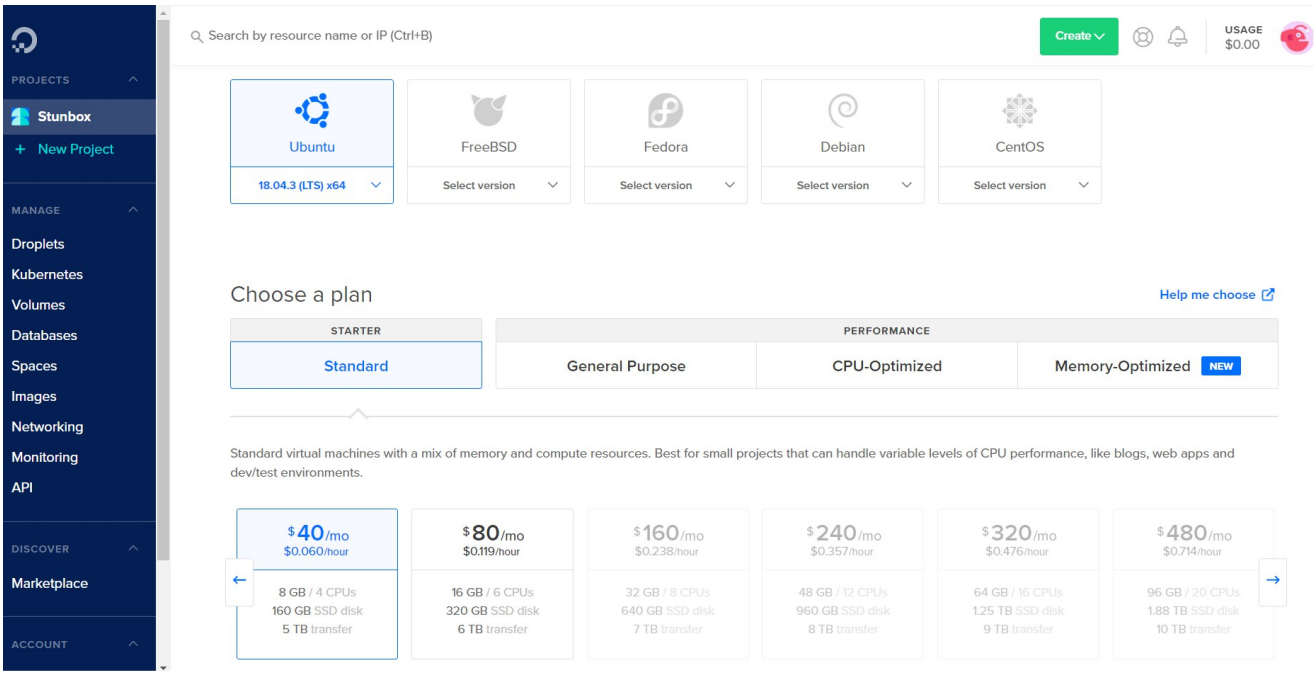
# Creación del Droplet

Vamos a utilizar :


para la creación del Droplet.

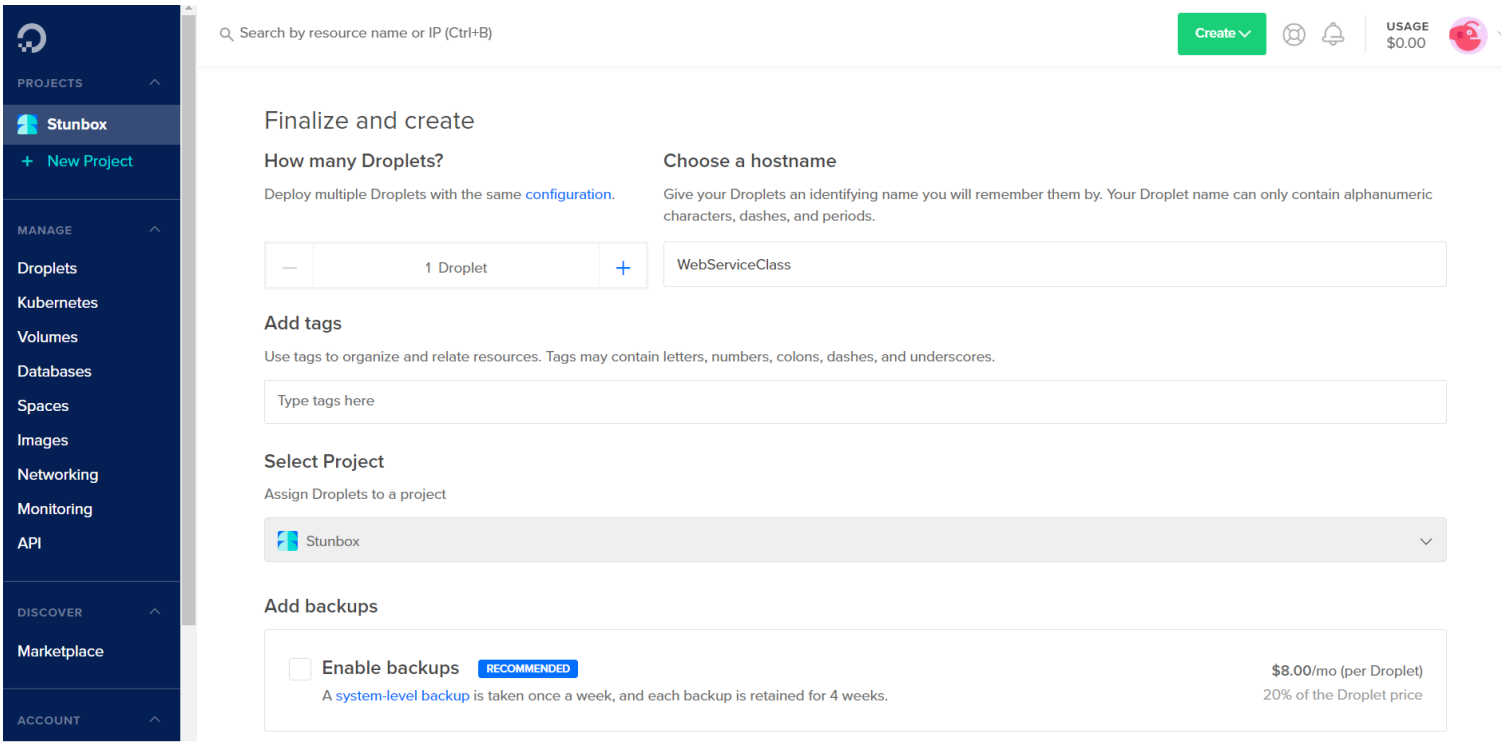
## 1._Adquirir Droplet en alguna plataforma

Lo primero que vamos a hacer es presionar en crear un nuevo Droplet, y nos va a llevar a esta sección, y vamos a seleccionar la plataforma a desear, en mi caso es el ubuntu, plan estándar , costos y demás.



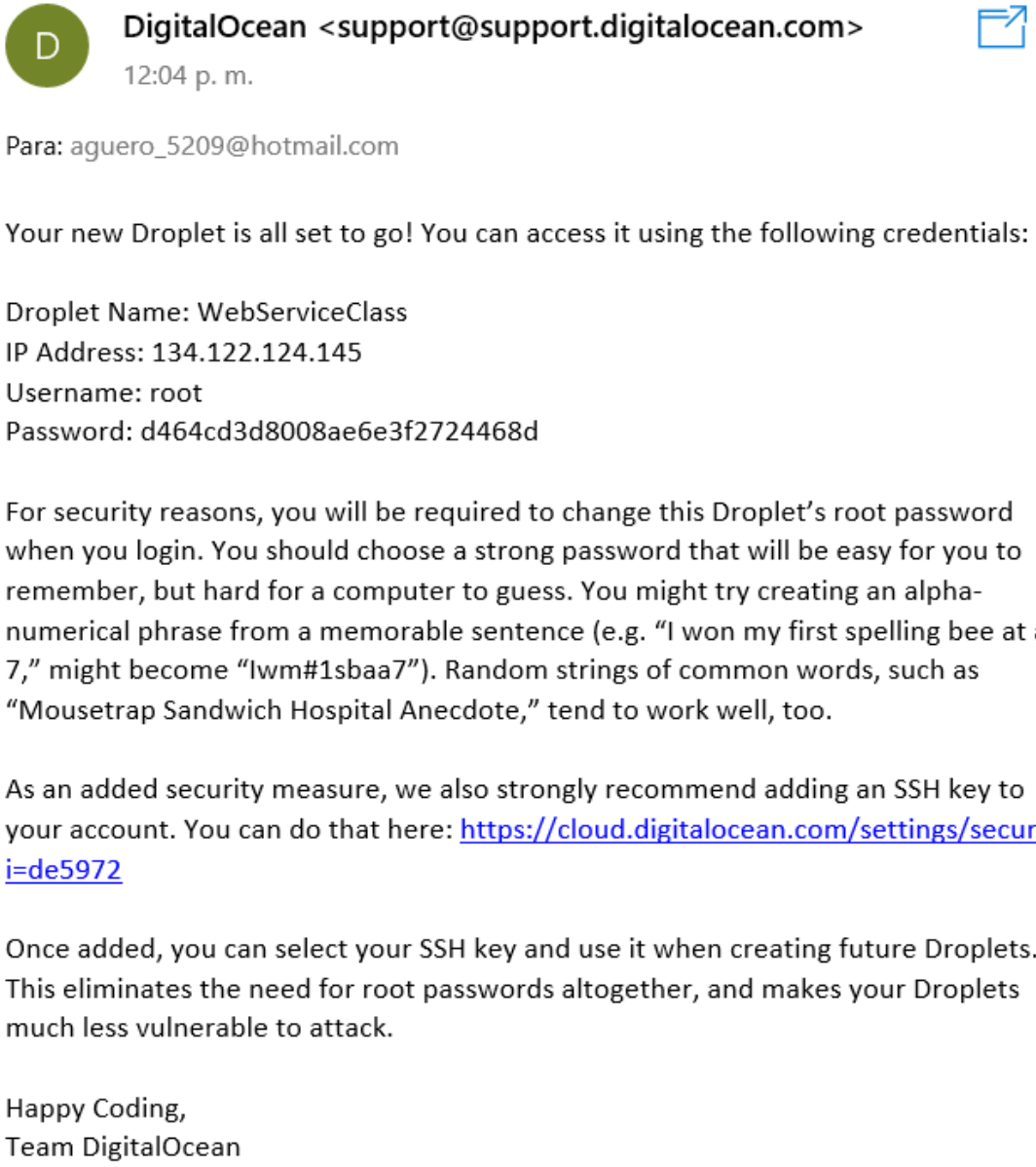

También ponemos la opción que esta justo abajo de este screen que es la de One-time password, en el cual nos enviaran por correo la contraseña y otros datos que utilizaremos.

Elegimos un nombre para nuestro host, el proyecto al que lo queremos adjuntar y lo creamos!



## 2._Acceder vía ssh

Si revisamos nuestro correo, nos habrán mandado nuestros datos junto con la contraseña y ip.



D

**DigitalOcean <support@support.digitalocean.com>**

12:04 p. m.

Para: aguero_5209@hotmail.com

Your new Droplet is all set to go! You can access it using the following credentials:

Droplet Name: WebServiceClass
IP Address: 134.122.124.145
Username: root
Password: d464cd3d8008ae6e3f2724468d

For security reasons, you will be required to change this Droplet's root password when you login. You should choose a strong password that will be easy for you to remember, but hard for a computer to guess. You might try creating an alpha-numerical phrase from a memorable sentence (e.g. "I won my first spelling bee at age 7," might become "Iwm#1sbaa7"). Random strings of common words, such as "Mousetrap Sandwich Hospital Anecdote," tend to work well, too.

As an added security measure, we also strongly recommend adding an SSH key to your account. You can do that here: https://cloud.digitalocean.com/settings/security?i=de5972

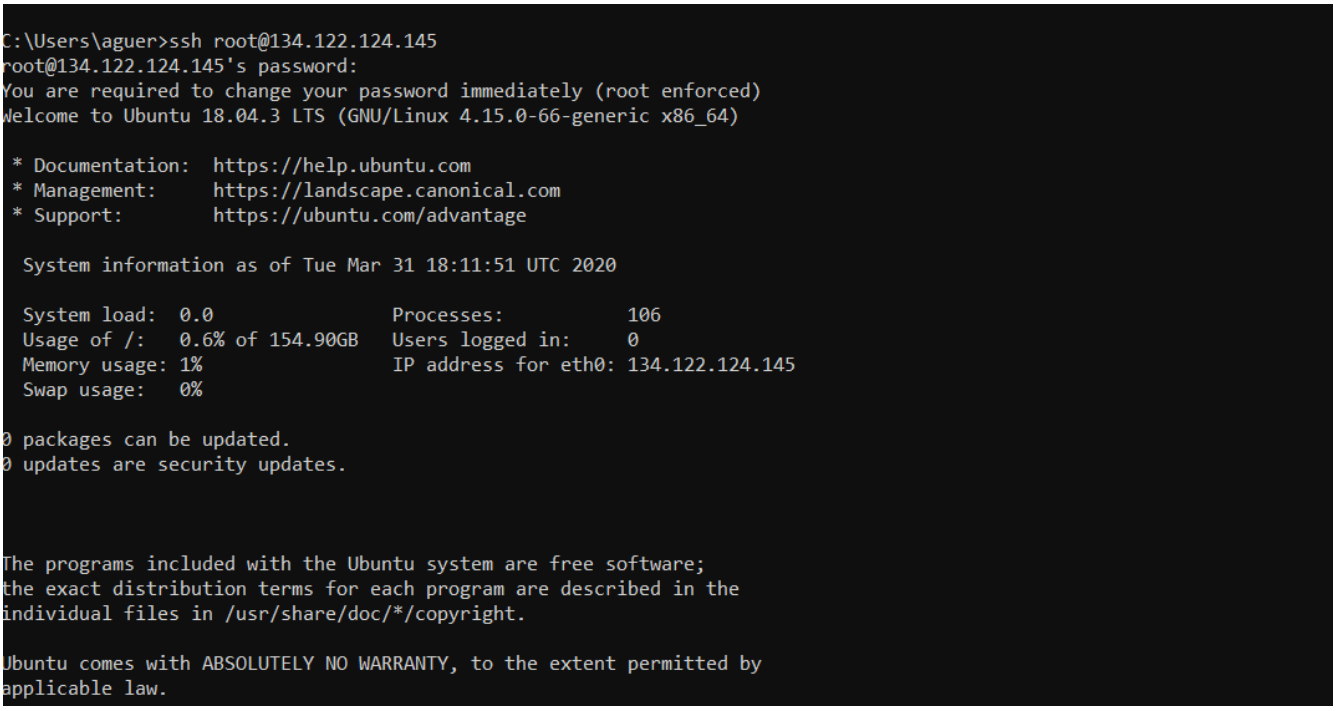
Once added, you can select your SSH key and use it when creating future Droplets. This eliminates the need for root passwords altogether, and makes your Droplets much less vulnerable to attack.

Happy Coding,
Team DigitalOcean

Ahora vamos a abrir nuestra terminal de Windows y vamos a ingresar unn comando el cual nos permitirá acceder de manera remota a nuestro sistema en ubuntu. → “ssh root@134.122.145 ”

**ssh** → comando para hacer la conexión

Estos 2 comandos llegan en el correo:
**root** → nombre de usuario
**134.122.145** → dirección ip del sistema



**2._Instalar Docker**

Lo primero que hay que hacer es desinstalar las versionas anteriores que tengamos.. y eso se hace con este comando:

“sudo apt-get remove docker docker-engine docker.io containerd runc”

```
root@WebServiceClass:~# sudo apt-get remove docker docker-engine docker.io containerd runc
Reading package lists... Done
Building dependency tree
Reading state information... Done
E: Unable to locate package docker
E: Unable to locate package docker-engine
E: Unable to locate package docker.io
E: Couldn't find any package by glob 'docker.io'
E: Couldn't find any package by regex 'docker.io'
E: Unable to locate package containerd
E: Unable to locate package runc
```

Como es la primera vez instalando Docker no hay nada que desinstalar solo nos aseguramos, no te preocupes!

Ahora pondremos el sig comando :

“ sudo apt-get install \
apt-transport-https \
curl \
gnupg-agent \
software-properties-common ”

Con este comando instalamos los paquetes necesarios por medio de un repositorio https

```
root@WebServiceClass:~# sudo apt-get install \
>     apt-transport-https \
>     ca-certificates \
>     curl \
>     gnupg-agent \
>     software-properties-common
Reading package lists... Done
Building dependency tree
Reading state information... Done
ca-certificates is already the newest version (20180409).
ca-certificates set to manually installed.
curl is already the newest version (7.58.0-2ubuntu3.8).
curl set to manually installed.
The following package was automatically installed and is no longer required:
  grub-pc-bin
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  python3-software-properties
The following NEW packages will be installed:
  apt-transport-https gnupg-agent
The following packages will be upgraded:
  python3-software-properties software-properties-common
2 upgraded, 2 newly installed, 0 to remove and 115 not upgraded.
Need to get 40.2 kB of archives.
After this operation, 196 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

Ahora agregaremos la GPG key oficial de docker con este comando:

“ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-gey add -”

y con este la verificamos:

“ sudo apt-key fingerprint 0EBFCD88”

```
root@WebServiceClass:~# curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
OK
root@WebServiceClass:~# sudo apt-key fingerprint 0EBFCD88
pub   rsa4096 2017-02-22 [SCEA]
      9DC8 5822 9FC7 DD38 854A  E2D8 8D81 803C 0EBF CD88
uid           [ unknown] Docker Release (CE deb) <docker@docker.com>
sub   rsa4096 2017-02-22 [S]
```

Ahora establecemos correctamente el repositorio para ubuntu con este comando:

“ sudo add-apt-repository \
“deb [arch=amd64] https://download.docker.com/linux/ubuntu \
$(lsb_release -cs) \
stable” ”

```
root@WebServiceClass:~# sudo add-apt-repository \
>    "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
>    $(lsb_release -cs) \
>    stable"
Get:1 https://download.docker.com/linux/ubuntu bionic InRelease [64.4 kB]
Get:2 http://mirrors.digitalocean.com/ubuntu bionic InRelease [242 kB]
Hit:3 http://security.ubuntu.com/ubuntu bionic-security InRelease
Get:4 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages [11.0 kB]
Hit:5 http://mirrors.digitalocean.com/ubuntu bionic-updates InRelease
Hit:6 http://mirrors.digitalocean.com/ubuntu bionic-backports InRelease
Fetched 318 kB in 1s (484 kB/s)
Reading package lists... Done
```

Ahora si, una vez preparado todo comenzaremos con los comandos para instalar docker.

Primero que nada aplicamos un update:

“ sudo apt-get update ”

Vamos a instalar la ultima versión de docker con este comando:

“ sudo apt-get install docker-ce docker-ce-cli containerd.io ”

Asi:

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

Listo!
Ahora vamos a verificar que el Docker se haya instalado correctamente!
Vamos a correr este comando:

“ sudo docker run hello-world ” y después “ docker ps ”
solo para estar seguros…

```
root@WebServiceClass:~# sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
1b930d010525: Pull complete
Digest: sha256:f9dfddf63636d84ef479d645ab5885156ae030f611a56f3a7ac7f2fdd86d7e4e
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/

root@WebServiceClass:~# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS
NAMES
```

Ahora vamos a instalar el Docker-compose!

Solo vamos a ingresar estos 3 comandos consecutivos para que ya quede!

Descargamos el ultimo realease de Docker compose:

“ 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 ”

Aplicamos los permisos correspondientes:

“sudo chmod +x /usr/local/bin/docker-compose”

Sacamos el as bajo la mandga con este comando(comando extra):

“ sudo ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose ”

Por ultimo simple y sencillamente verificamos la version con:

“ docker-compose --version”

```
root@WebServiceClass:~# 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  Speed
100   617  100   617    0     0  11425      0 --:--:-- --:--:-- --:--:-- 11425
100 16.3M  100 16.3M    0     0  60.6M      0 --:--:-- --:--:-- --:--:-- 60.6M
root@WebServiceClass:~# sudo chmod +x /usr/local/bin/docker-compose
root@WebServiceClass:~# sudo ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose
root@WebServiceClass:~# docker-compose --version
docker-compose version 1.25.4, build 8d51620a
```

**3.Clonar Repositorio**

Lo primero que vamos a hacer en posicionarnos en nuestra carpeta home con : cd/home

Ahora vamos a clonar nuestro repositorio con nuestros archivos, en este caso mi repositorio esta en github, usaremos este comando:

“ git clone https://github.com/Stunbox/Devops.git ”

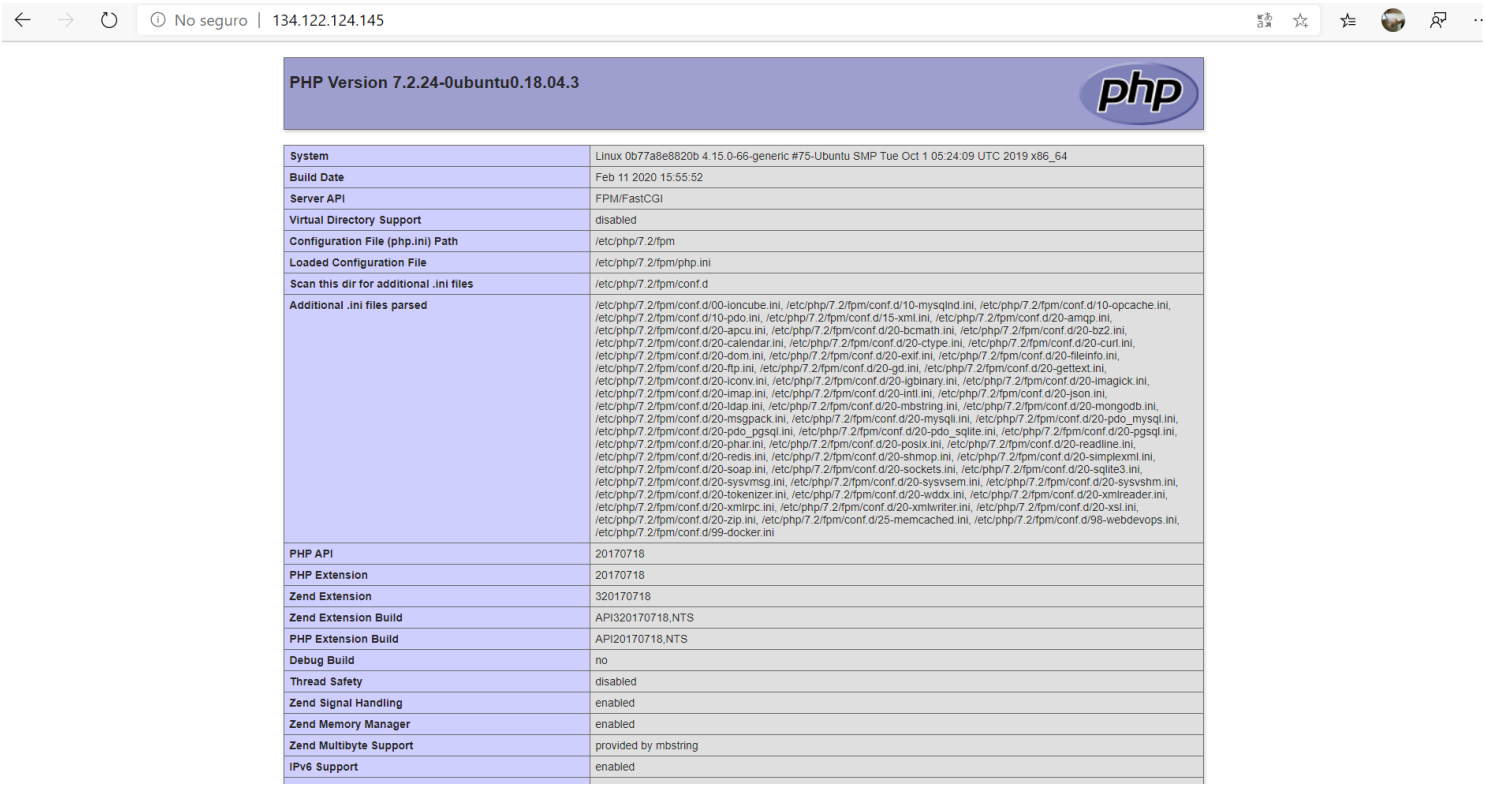
Aquí estoy usando el link a mi repositorio pero puede ir cualquier otro!

```
root@WebServiceClass:/home# git clone https://github.com/Stunbox/Devops.git
Cloning into 'Devops'...
remote: Enumerating objects: 107, done.
remote: Counting objects: 100% (107/107), done.
remote: Compressing objects: 100% (40/40), done.
remote: Total 107 (delta 61), reused 96 (delta 59), pack-reused 0
Receiving objects: 100% (107/107), 9.81 MiB | 10.00 MiB/s, done.
Resolving deltas: 100% (61/61), done.
```

Todo listo? Ahora vamos a levantar el contenedor!

```
root@WebServiceClass:/home/Devops# docker-compose up -d
```

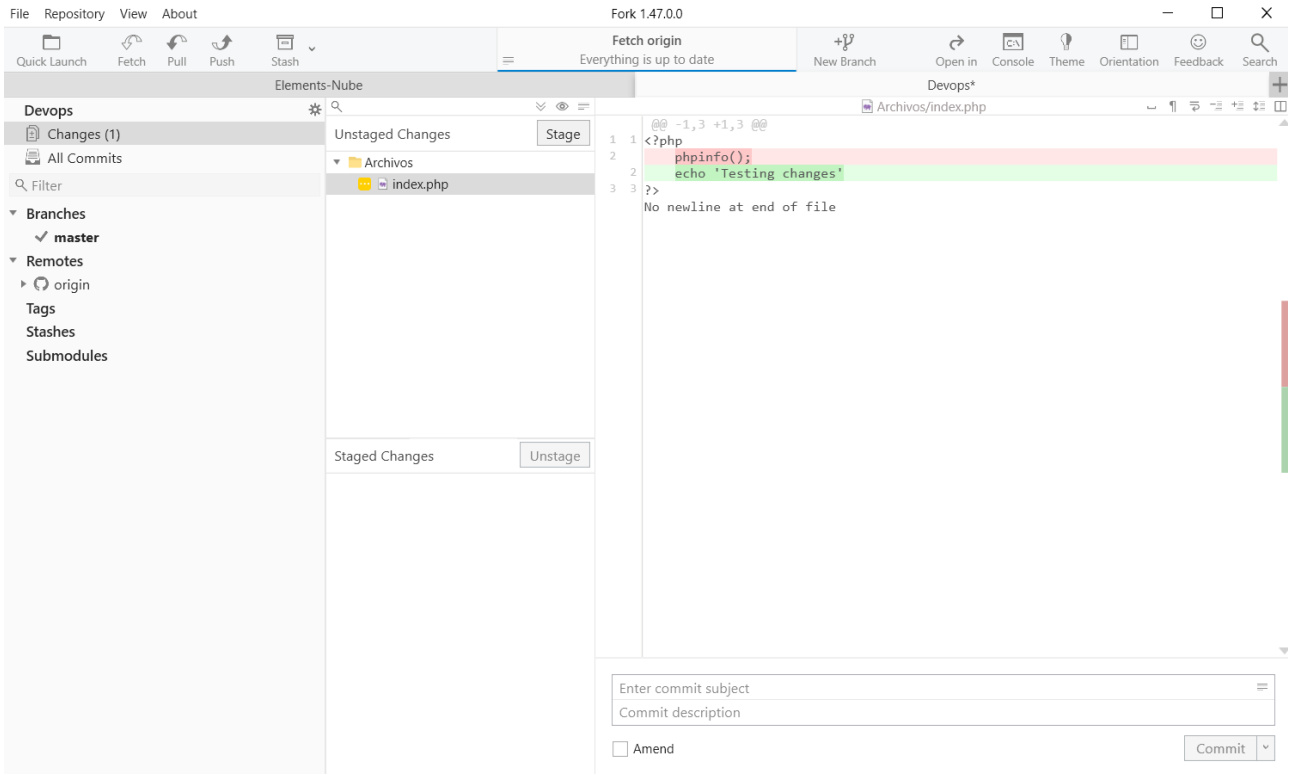
Vientos! Ahora vamos a acceder a nuestra pagina por medio de la ip que nos proporcionaron desde un inicio en el correo!




| | |
|---|---|
| System | Linux 0b77a8e8820b 4.15.0-66-generic #75-Ubuntu SMP Tue Oct 1 05:24.09 UTC 2019 x86_64 |
| Build Date | Feb 11 2020 15:55:52 |
| Server API | FPM/FastCGI |
| Virtual Directory Support | disabled |
| Configuration File (php.ini) Path | /etc/php/7.2/fpm |
| Loaded Configuration File | /etc/php/7.2/fpm/php.ini |
| Scan this dir for additional .ini files | /etc/php/7.2/fpm/conf.d |
| Additional .ini files parsed | /etc/php/7.2/fpm/conf.d/00-ioncube.ini, /etc/php/7.2/fpm/conf.d/10-mysqlnd.ini, /etc/php/7.2/fpm/conf.d/10-opcache.ini, /etc/php/7.2/fpm/conf.d/10-pdo.ini, /etc/php/7.2/fpm/conf.d/15-xml.ini, /etc/php/7.2/fpm/conf.d/20-amqp.ini, /etc/php/7.2/fpm/conf.d/20-apcu.ini, /etc/php/7.2/fpm/conf.d/20-bcmath.ini, /etc/php/7.2/fpm/conf.d/20-bz2.ini, /etc/php/7.2/fpm/conf.d/20-calendar.ini, /etc/php/7.2/fpm/conf.d/20-ctype.ini, /etc/php/7.2/fpm/conf.d/20-curl.ini, /etc/php/7.2/fpm/conf.d/20-dom.ini, /etc/php/7.2/fpm/conf.d/20-exif.ini, /etc/php/7.2/fpm/conf.d/20-fileinfo.ini, /etc/php/7.2/fpm/conf.d/20-ftp.ini, /etc/php/7.2/fpm/conf.d/20-gd.ini, /etc/php/7.2/fpm/conf.d/20-gettext.ini, /etc/php/7.2/fpm/conf.d/20-iconv.ini, /etc/php/7.2/fpm/conf.d/20-igbinary.ini, /etc/php/7.2/fpm/conf.d/20-imagick.ini, /etc/php/7.2/fpm/conf.d/20-imap.ini, /etc/php/7.2/fpm/conf.d/20-intl.ini, /etc/php/7.2/fpm/conf.d/20-json.ini, /etc/php/7.2/fpm/conf.d/20-ldap.ini, /etc/php/7.2/fpm/conf.d/20-mbstring.ini, /etc/php/7.2/fpm/conf.d/20-mongodb.ini, /etc/php/7.2/fpm/conf.d/20-msgpack.ini, /etc/php/7.2/fpm/conf.d/20-mysqli.ini, /etc/php/7.2/fpm/conf.d/20-pdo_mysql.ini, /etc/php/7.2/fpm/conf.d/20-pdo_pgsql.ini, /etc/php/7.2/fpm/conf.d/20-pdo_sqlite.ini, /etc/php/7.2/fpm/conf.d/20-pgsql.ini, /etc/php/7.2/fpm/conf.d/20-phar.ini, /etc/php/7.2/fpm/conf.d/20-posix.ini, /etc/php/7.2/fpm/conf.d/20-readline.ini, /etc/php/7.2/fpm/conf.d/20-redis.ini, /etc/php/7.2/fpm/conf.d/20-shmop.ini, /etc/php/7.2/fpm/conf.d/20-simplexml.ini, /etc/php/7.2/fpm/conf.d/20-soap.ini, /etc/php/7.2/fpm/conf.d/20-sockets.ini, /etc/php/7.2/fpm/conf.d/20-sqlite3.ini, /etc/php/7.2/fpm/conf.d/20-sysvmsg.ini, /etc/php/7.2/fpm/conf.d/20-sysvsem.ini, /etc/php/7.2/fpm/conf.d/20-sysvshm.ini, /etc/php/7.2/fpm/conf.d/20-tokenizer.ini, /etc/php/7.2/fpm/conf.d/20-wddx.ini, /etc/php/7.2/fpm/conf.d/20-xmlreader.ini, /etc/php/7.2/fpm/conf.d/20-xmlrpc.ini, /etc/php/7.2/fpm/conf.d/20-xmlwriter.ini, /etc/php/7.2/fpm/conf.d/20-xsl.ini, /etc/php/7.2/fpm/conf.d/20-zip.ini, /etc/php/7.2/fpm/conf.d/25-memcached.ini, /etc/php/7.2/fpm/conf.d/98-webdevops.ini, /etc/php/7.2/fpm/conf.d/99-docker.ini |
| PHP API | 20170718 |
| PHP Extension | 20170718 |
| Zend Extension | 320170718 |
| Zend Extension Build | API320170718,NTS |
| PHP Extension Build | API20170718,NTS |
| Debug Build | no |
| Thread Safety | disabled |
| Zend Signal Handling | enabled |
| Zend Memory Manager | enabled |
| Zend Multibyte Support | provided by mbstring |
| IPv6 Support | enabled |

Un poco aburrida mi pagina pero por ahora solo estamos probando!

Ahora vamos a realizar un cambio para probar que si funciona




Hacemos push y en nuestro consola hacemos un pull para bajar todas las actualizaciones con este comando:

“git pull origin master”

Y. . .


Listo! Nuestros cambios son visualizados de forma exitosa!