



Integración Continua, DevOps, Docker + Flask.

steven sierra forero | github.com/bizoru | ingenieria.sierra@gmail.com



Hi All!

Soy Steven Sierra Forero.

Desarrollador de Software apasionado por el software libre, he programado en C, C#, C++, VBA, PHP, JSP, Java, Javascript, Haskell, Node, Logo, Python, Ruby, Golang, bash, perl y esta es mi cuenta de Github:

`github.com/bizoru`

Ahora en el camino de DevOps e integración continua.

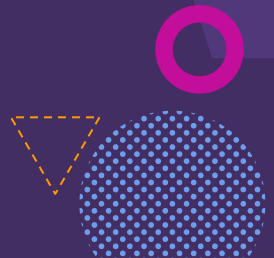




1.

Integración Continua

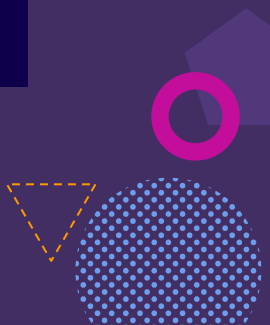

En términos coloquiales.





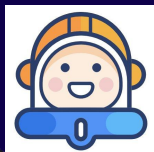
“

Es básicamente una práctica de desarrollo de software, donde el desarrollador sube el código fuente a un repositorio compartido donde un sistema automatizado lo compila o construye y produce artefactos de software si todo salió bien.



“

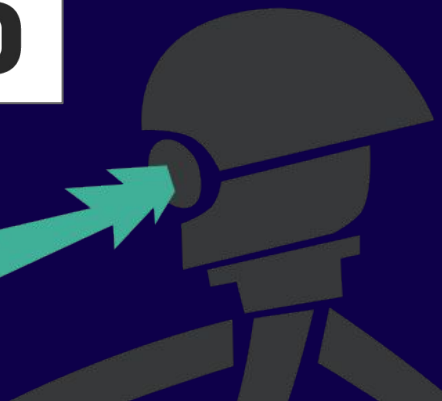
Herramientas usuales en el ámbito de integración continua.



Bamboo



Travis CI





2.

DevOps

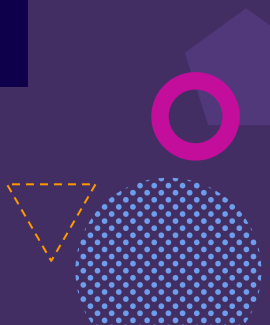

En términos coloquiales.





“

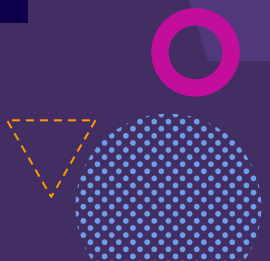

Es la participación en conjunto de ingenieros de desarrollo e ingenieros de operaciones en el ciclo de desarrollo de software desde el diseño hasta el desarrollo llegando hasta el soporte de producción.



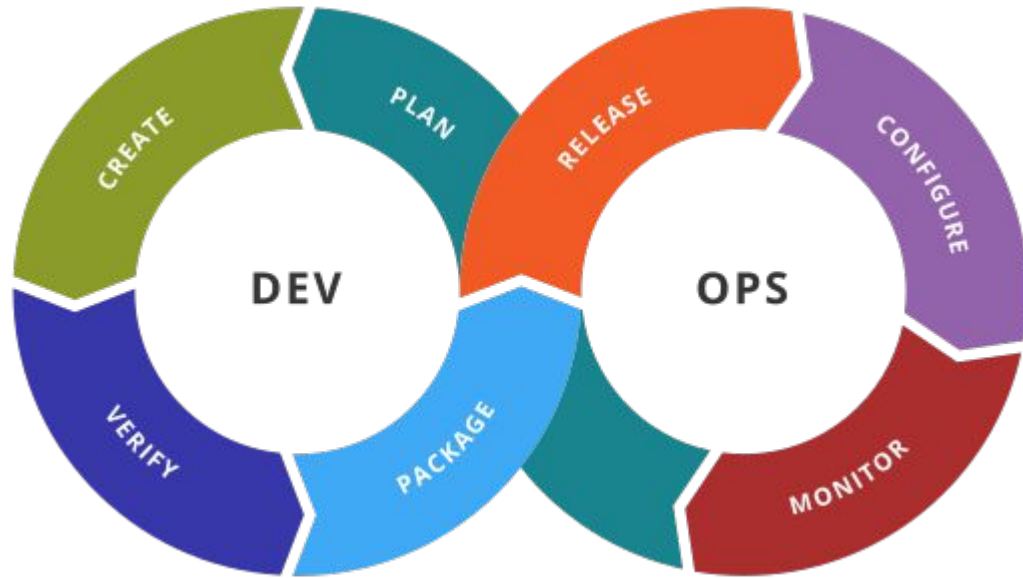


“

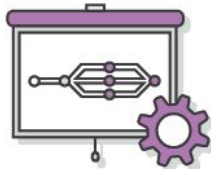
Puede llegar a ser complicado llegar a una definición que demarque claramente hasta dónde va el rol de un DevOps.



DevOps Toolchain



Fuente: Wikipedia.



Automatice el proceso de publicación de software

La entrega continua permite al equipo crear, probar y preparar automáticamente los cambios en el código para su envío a producción, con lo que se mejora la eficacia y rapidez de la entrega de software.

“

Ventajas



Encuentre y arregle los errores con mayor rapidez

Su equipo puede descubrir y arreglar los errores antes de que se conviertan en problemas más graves gracias a las pruebas más frecuentes y exhaustivas. La entrega continua le permite realizar tipos de pruebas adicionales en el código con facilidad, ya que todo el proceso se ha automatizado.

“

Ventajas

3. Docker

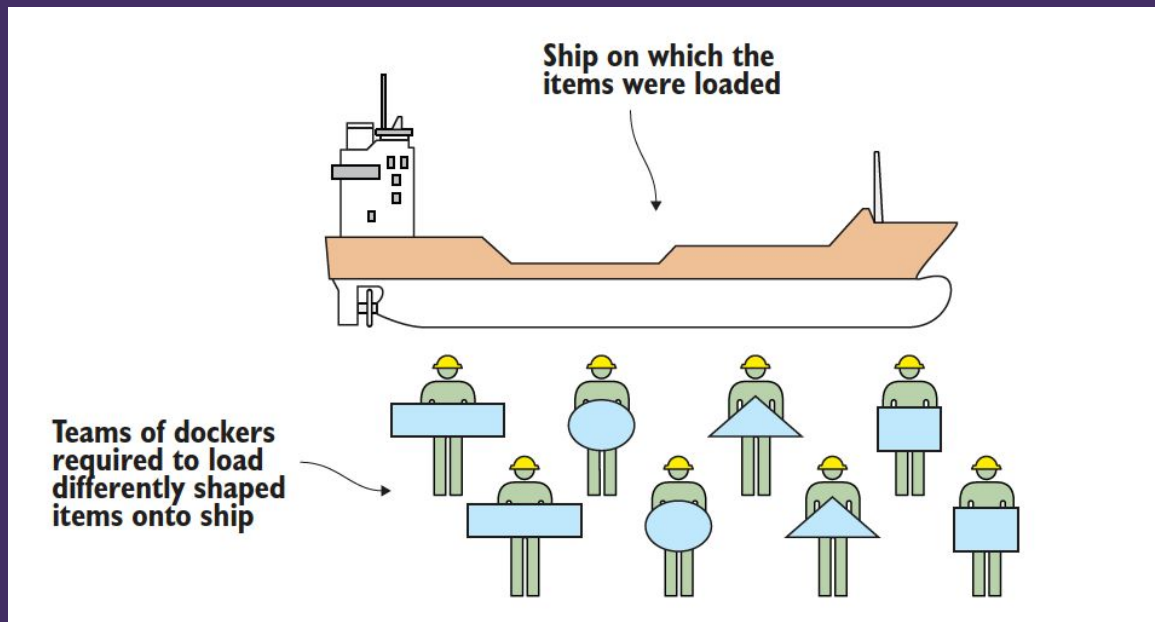
En términos coloquiales.



“

Plataforma de software que permite empaquetar software sin necesidad de depender de librerías o binarios del sistema. No es una máquina virtual es un contenedor.

Docker, entendiendo el nuevo paradigma.



Tomado del libro: Docker in Practice.

Docker, entendiendo el nuevo paradigma.

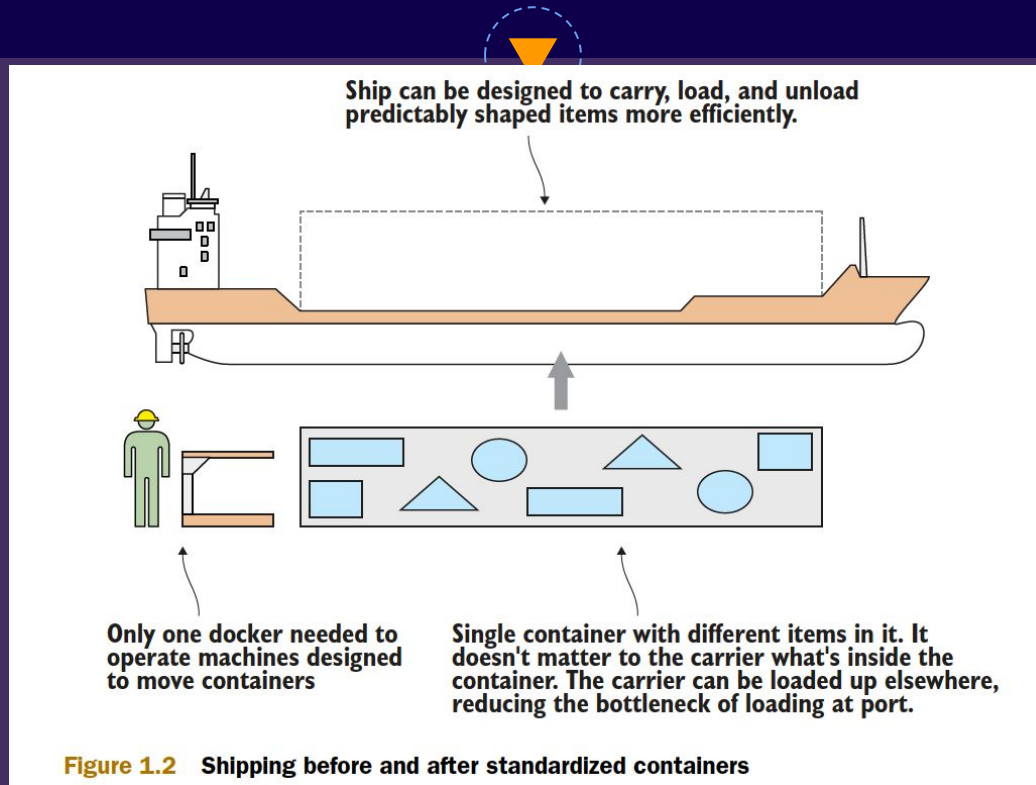
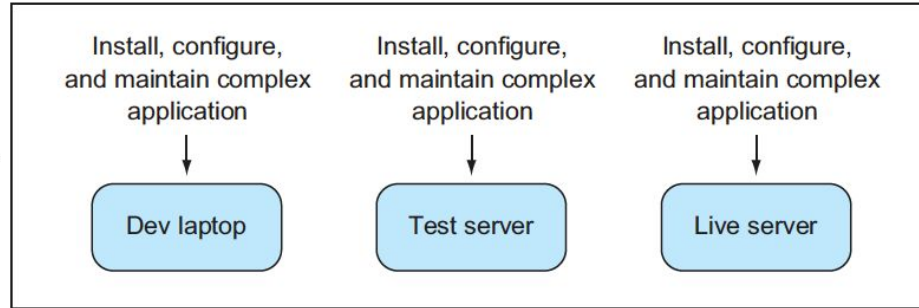


Figure 1.2 Shipping before and after standardized containers

La vida antes de Docker ...

**Three times the
effort to manage
deployment**

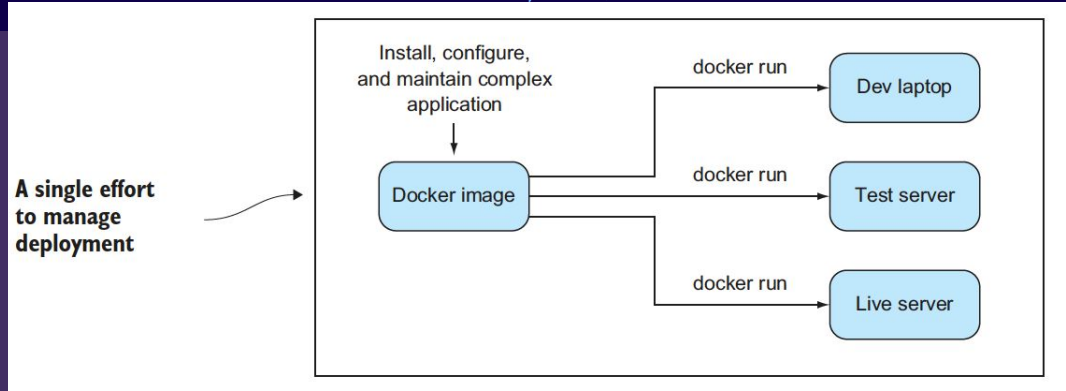
Life before Docker



Tomado del libro: Docker in Practice.

La vida después de Docker ...

Tomado del libro: Docker in Practice.

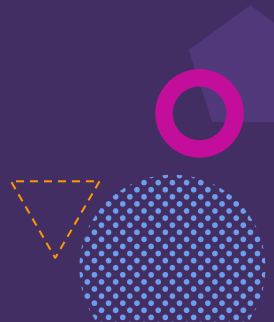




4.

Flask

En términos coloquiales.



“

Es un conjunto de librerías en python que permiten desarrollar aplicaciones web. Con estas se puede crear un blog, una wiki o un sitio web comercial.



Flask

web development,
one drop at a time

[overview](#) // [docs](#) // [community](#) // [snippets](#) // [extensions](#)

Flask is a microframework for Python based on Werkzeug, Jinja 2 and good intentions. And before you ask: It's [BSD licensed](#)!

Flask is Fun

Latest Version: [0.12](#)

```
from flask import Flask
app = Flask(__name__)

@app.route("/")
def hello():
    return "Hello World!"

if __name__ == "__main__":
    app.run()
```

And Easy to Setup

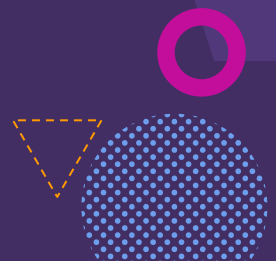
```
$ pip install Flask
$ python hello.py
* Running on http://localhost:5000/
```



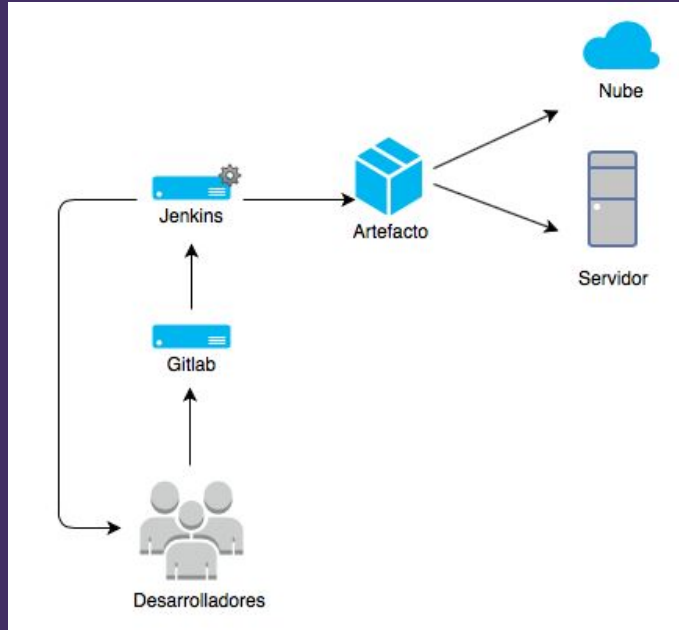
5.

A trabajar!

...



Escenario Básico de CI



1. El desarrollador crea un feature.
2. El feature sube a github o cualquier tipo de repositorio.
3. La herramienta de CI (Jenkins, Bamboo, Travis, Buddy, etc) ejecuta los comandos para tener un build estable.
4. Se produce un artefacto.
5. El artefacto se puede desplegar en un servidor, en la nube o en cualquier lugar.

Comandos Básicos de Docker.



- El primero de todos

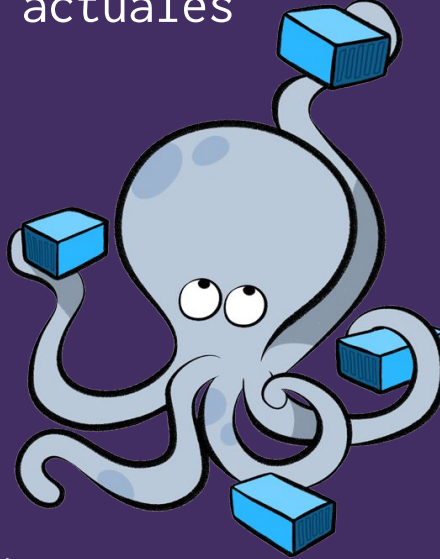


```
docker run hello-world
```

Comandos Básicos de Docker.



- Listar las imágenes actuales



docker images

Comandos Básicos de Docker.



- Listar las imágenes actuales

`docker images`

Comandos Básicos de Docker.



- Listar los containers que estan corriendo

```
docker ps -a
```

Comandos Básicos de Docker.



- Inspeccionar containers corriendo

```
docker inspect container
```

Comandos Básicos de Docker.



- Ver logs del container

```
docker logs container
```

Comandos Básicos de Docker.



- Eliminar Imagenes

```
docker rmi imagen
```

- Eliminar Containers

```
docker rm container
```

```
docker rm $(docker ps -aq)
```

Comandos Básicos de Docker.



- Traer una imagen del registry

```
docker pull imagen
```

- Parar, Arrancar un contenedor

```
docker stop container
```

```
docker start container
```

Comandos Básicos de Docker.



- Flags para el inicio de containers docker.

-p `host_port:docker_port`

-v `host_volume:docker_volume`

-d `detached mode`

Comandos Básicos de Docker.



- Flags para el inicio de containers docker.

`--name container_name`

`-i interactive`

`-t attach a terminal`

Comandos Básicos de Docker.



- Ejecutar un comando en un container corriendo

`docker exec flags container command`





Jenkins



- Ejecutar un comando en un container corriendo

```
docker run -d --name jenkins -p 8080:8080 -p 50000:50000 jenkins
```



Gitlab SCM



```
docker run --detach \  
  --hostname gitlab.example.com \  
  --publish 443:443 --publish 80:80 --publish 22:22 \  
  --name gitlab \  
  --restart always \  
  --volume $HOME/Downloads/gitlab-server/config:/etc/gitlab \  
  --volume $HOME/Downloads/gitlab-server/logs:/var/log/gitlab \  
  --volume $HOME/Downloads/gitlab-server/data:/var/opt/gitlab \  
  gitlab/gitlab-ce:latest
```

Servidor FTP



```
docker run -d --name ftpd_server -p 21:21  
-p 30000-30009:30000-30009  
-e "PUBLICHOST=localhost" stilliard/pure-ftpd:hardened
```

Fuentes de Consulta



https://hub.docker.com/_/jenkins/

<http://flask.pocoo.org>

<https://docs.gitlab.com/omnibus/docker/>

<https://hub.docker.com/r/stilliard/pure-ftpd/>



Hello World

CI

The big picture



Hello World

CI

The big picture



Gracias!

Alguna pregunta?

You can find me at github.com/bizoru &
ingenieria.sierra@gmail.com