

A person in winter gear and a large backpack is climbing a steep, snowy mountain slope. They are using ice axes and crampons. In the background, three other climbers are visible further up the slope. The sky is a clear, pale blue.

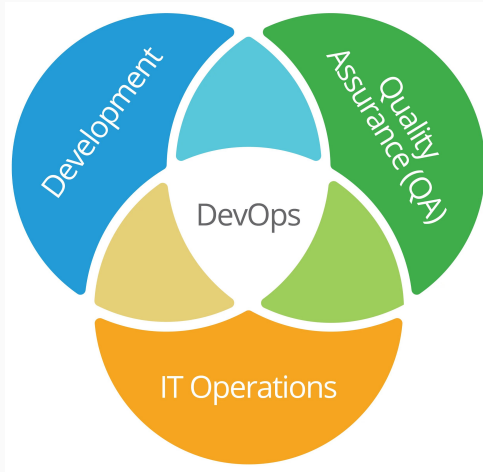
DevOps

@wjma90



- ¿Qué es DevOps?
- ¿Por qué DevOps?
- Cultura Organizativa
- Mundo Agile - Principios y Conceptos a DevOps
- Herramientas Agile vs DevOps. ¿Son complementos?
- Introducción a Docker
- Uso Básico de Docker

¿Qué es DevOps?



Se ha hablado mucho sobre qué es y qué no es DevOps, pero el consenso reciente es que ***DevOps se trata principalmente de cultura***. Las organizaciones que han adoptado esta cultura valoran la colaboración, la experimentación y el aprendizaje. En una cultura DevOps, todos los participantes en todas las fases del ciclo de vida de entrega de software se alinean en torno a un objetivo compartido: la entrega rápida de software estable y de alta calidad desde el concepto hasta el cliente.

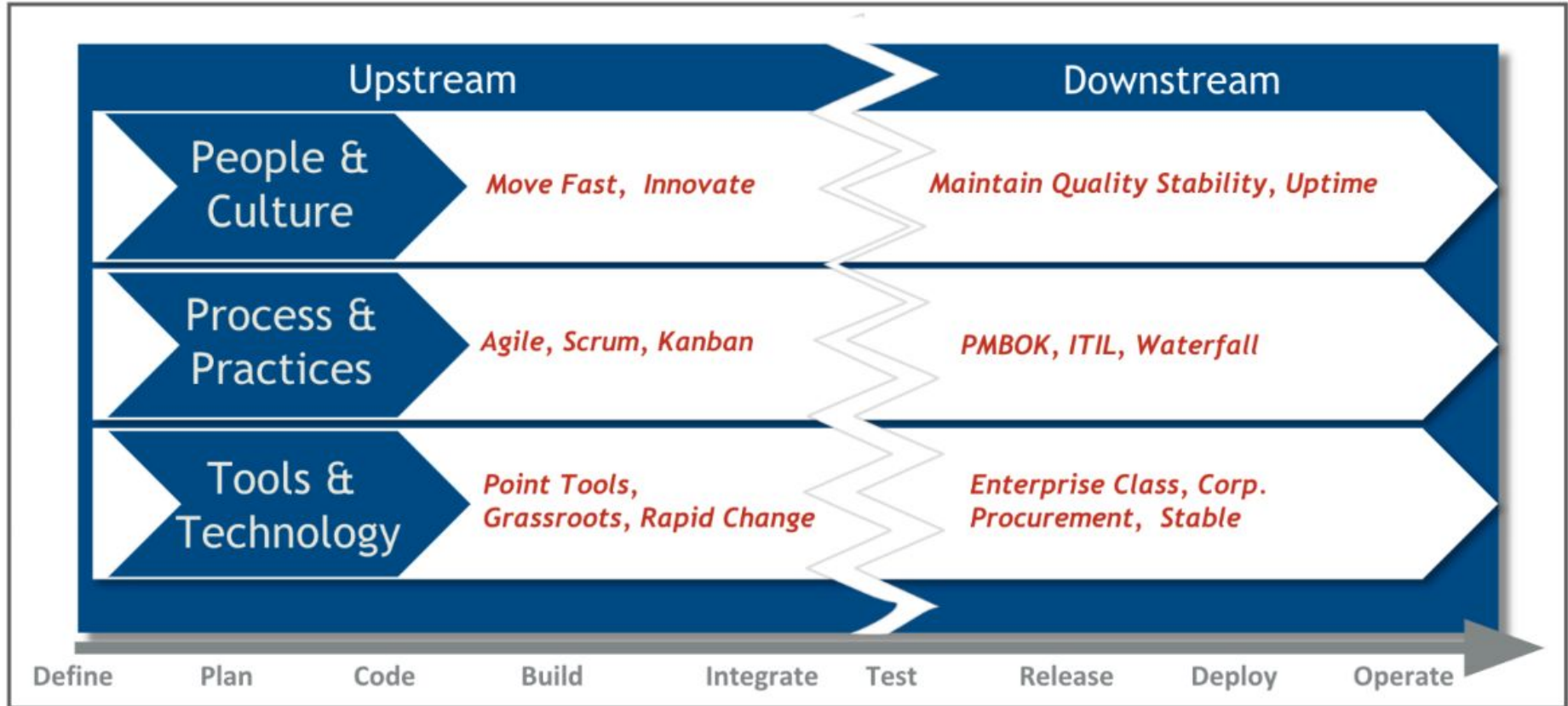
¿Por qué DevOps?

En pocas palabras, **las empresas que han adoptado DevOps están mejor posicionadas para consolidar su posición** en los mercados existentes y expandirse a otros nuevos. De hecho, la adopción de DevOps se está convirtiendo rápidamente en el estándar de facto para el desarrollo de software. Un vistazo rápido a los ejemplos de la industria.

Una vez que se implementa DevOps, **las empresas han visto un aumento en la satisfacción de los interesados en el proceso de desarrollo**. Debido a que están aplicando prácticas modernas de desarrollo de software, estas compañías están atrayendo y reteniendo mejores talentos, quienes impulsarán nuevas mejoras de calidad, mejoras de productividad y finalmente el éxito en la compañía.



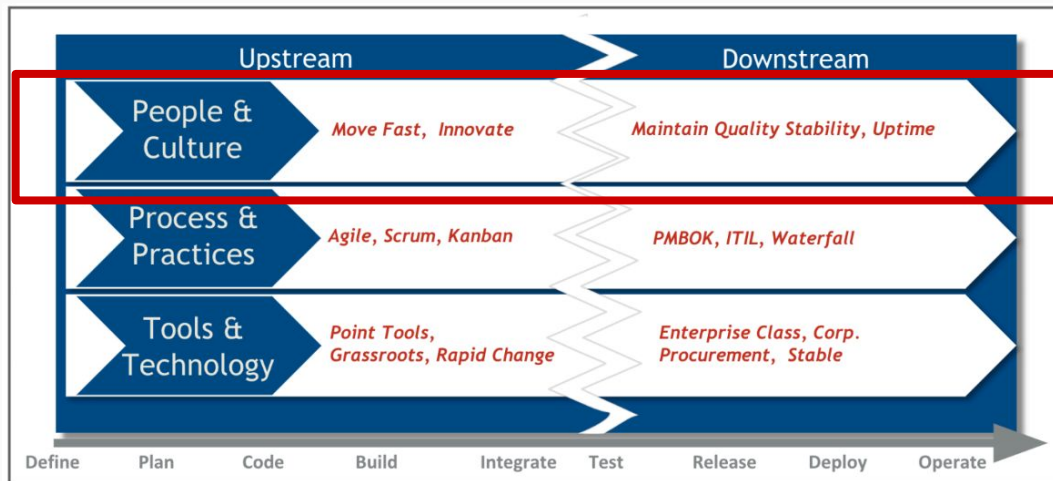
Cultura Organizativa - Trinidad DevOps



Cultura Organizativa - Trinidad DevOps

La cultura de **desarrollo generalmente prioriza la velocidad y la innovación**, mientras que en sentido descendente, la cultura de **operaciones tiene la tarea de mantener la calidad, la estabilidad y el tiempo de actividad**. Upstream, el desarrollo utiliza herramientas puntuales para definir y construir software utilizando métodos ágiles. En sentido descendente, las herramientas de clase empresarial son la norma para administrar la prueba, el lanzamiento, la implementación y la operación del software

En este sentido, debemos preocuparnos en alinear ambas partes. Tener un flujo que permita al software transitar entre ambas partes sin causar conflictos

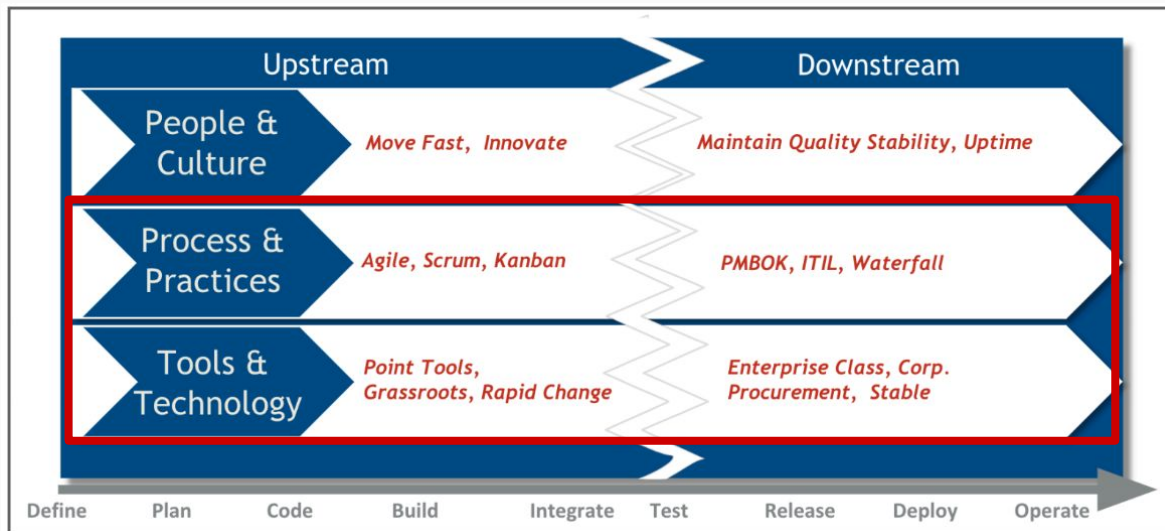


Cultura Organizativa - Trinidad DevOps

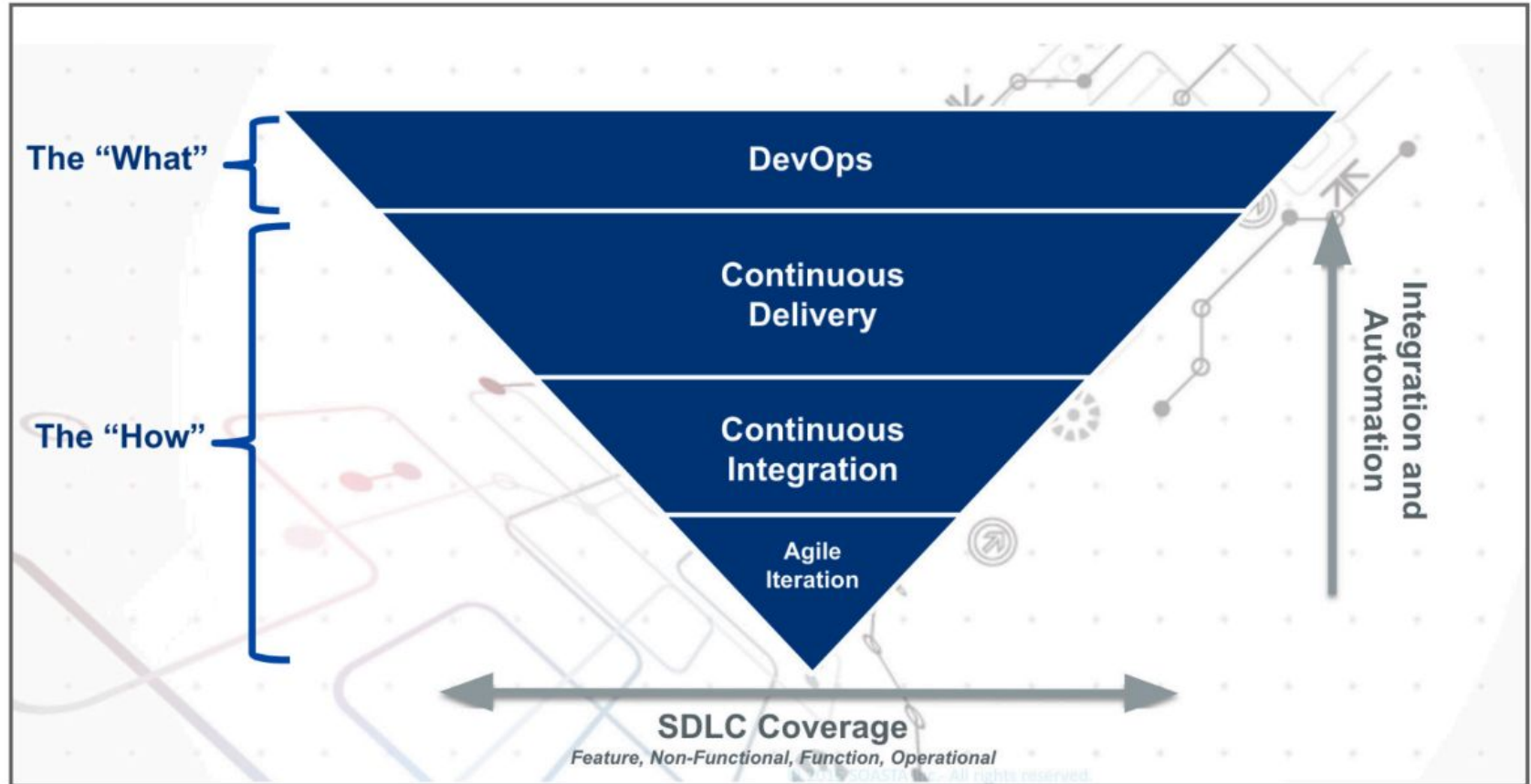
En los procesos y prácticas es mucho más probable que las reuniones posteriores se llenen de charlas sobre ITIL y PMBOK que Kanban y/o scrum. **DevOps se trata de conectar estos mundos y eliminar el abismo** que existe entre el río arriba y el río abajo.

A menudo, se pierde el tiempo sobre **las discusiones sobre cómo DevOps se centran demasiado en el núcleo técnico**

desde el momento en que los desarrolladores confirman el código hasta el momento en que se implementa el software en un servidor. **En realidad, DevOps se extiende desde el concepto inicial hasta la implementación del cliente**



Cultura Organizativa - Trinidad DevOps



La situación actual en el ciclo de desarrollo de software (SDLC) se resume en gaps entre las partes



Aplicar la cultura 'Agile' nos permite romper la brecha entre negocio y el equipo de desarrollo

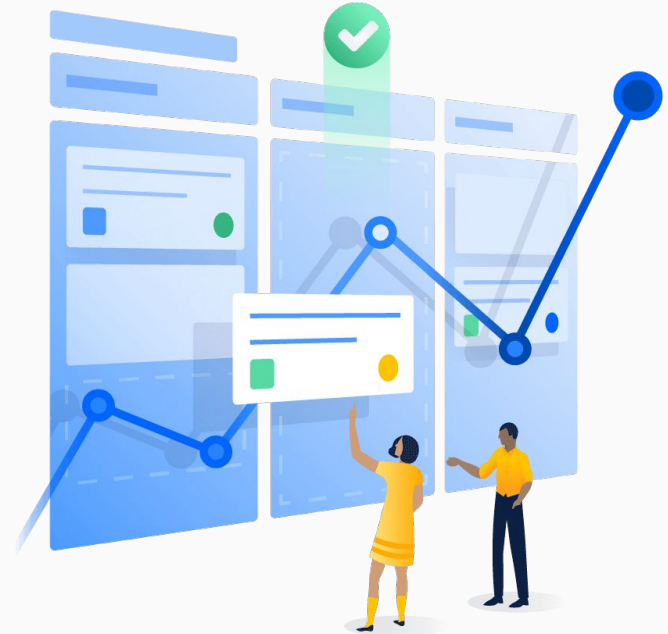


Agile - how?

SCRUM



KANBAN



Agile - Scrum

Roles

-  Scrum Master
-  Scrum Team
-  Product Owner

Events

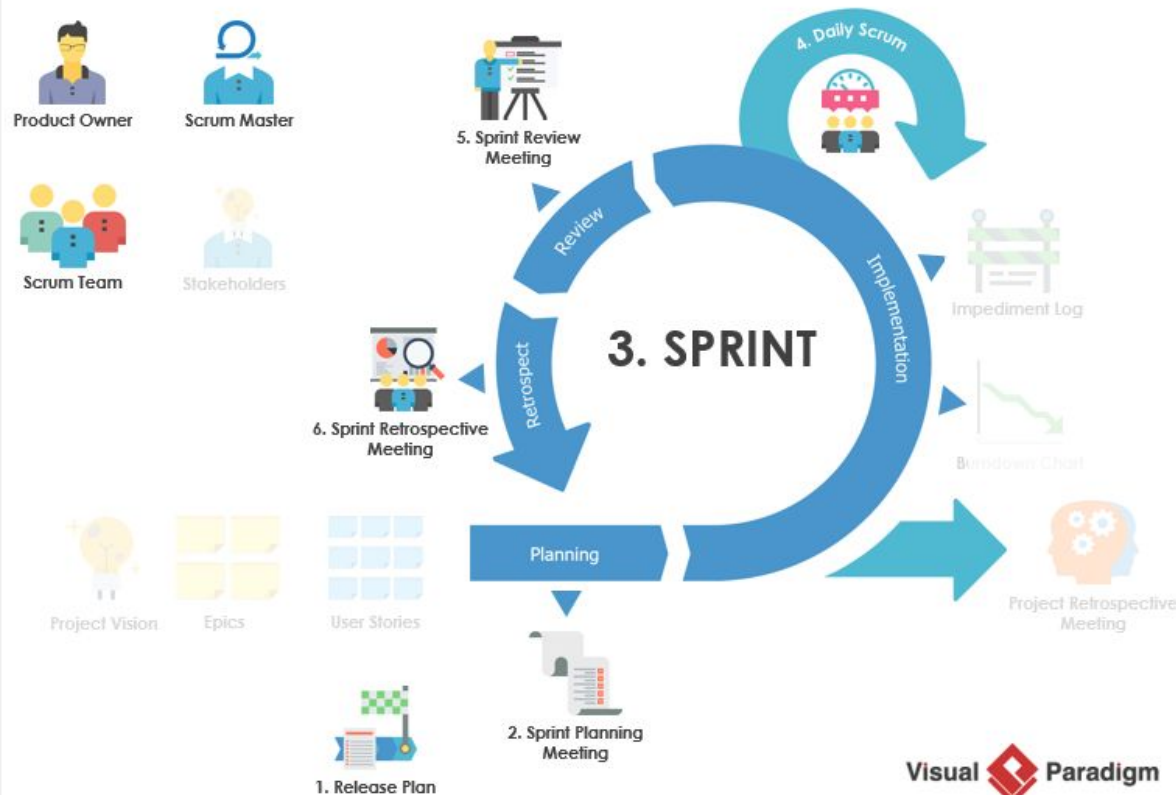
-  Release Planning Meeting
-  Sprint Planning Meeting
-  Sprint
-  Daily Scrum
-  Sprint Review
-  Sprint Retrospective

Artifacts

-  Product Backlog
-  Sprint Backlog
-  Sprint Burndown
-  Release Burndown

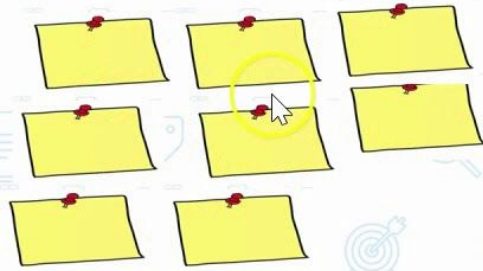
Rules

(bind the model together)

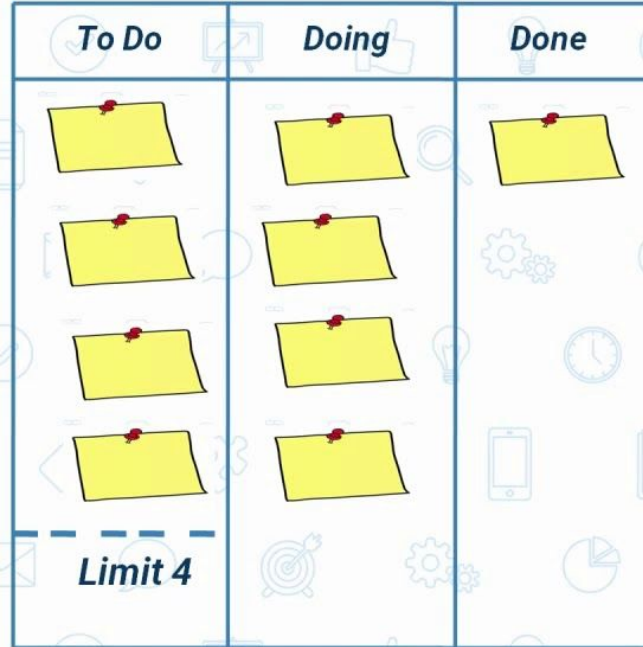
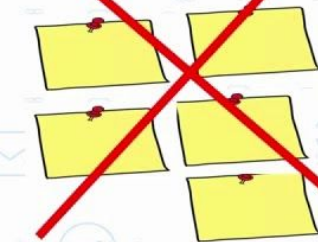


Kanban Board

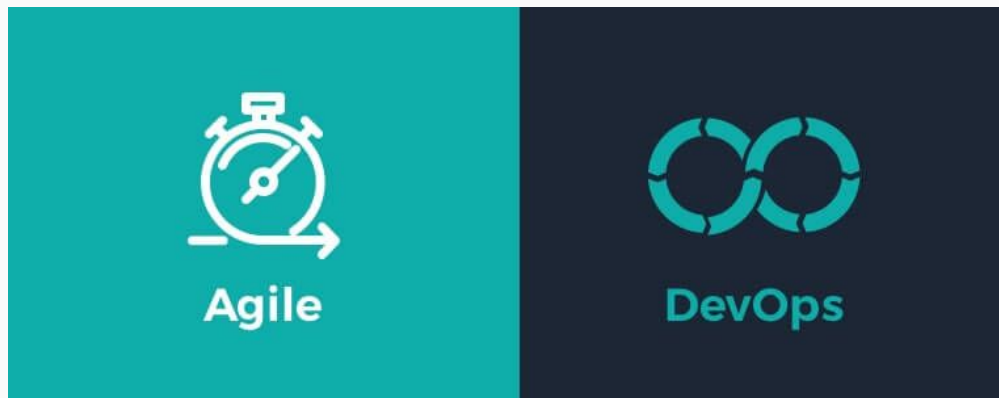
Product Backlog



Sprint Backlog



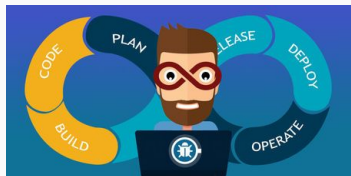
Agile - how?



**Mucha gente piensa que Agile significa Scrum y DevOps significa Entrega Continua
¡Esta simplificación excesiva crea una tensión innecesaria entre Agile y DevOps.
Ambos son complementos.**

*“Hasta que DevOps sea convencional, DevOps no será un resultado orgánico de Scrum.
Si el equipo contrata a un entrenador ágil o DevOps es probablemente irrelevante,
siempre y cuando esa persona pueda aportar experiencia en automatización en la
construcción, prueba e implementación de software”... Atlassian*

Agile - ¿Cómo DevOps puede ayudar?



JIRA Dashboards ▾ Projects ▾ Issues ▾ Boards ▾ Structure ▾ Create Search ? ⚙️ 🌐

Site Preparations Board

Backlog

QUICK FILTERS: Only My Issues Recently Updated

▼ Sprint 1 4 issues 25 10 15
31/Jul/13 5:18 PM • 14/Aug/13 5:18 PM Linked pages

- SP-6 Build access road Flatten Site 10
- SP-5 Relocate trees 2 Relocate inhabitants 15
- SP-1 Relocate Elves 2 Relocate inhabitants 5
- SP-2 Bulldoze French Alps Flatten Site 20

Backlog 5 issues Create Sprint

- SP-3 Remove waste rock and soil Flatten Site 6
- SP-8 Dig escape tunnels for staff Build Infrastructure 8
- SP-9 Build a transparent dome over the theme Build Infrastructure 9

Site Preparations / SP-1 Relocate Elves Estimate: 5

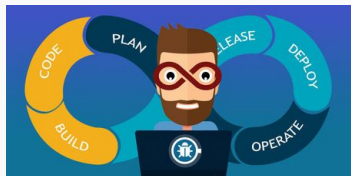
Structure

Portfolio Overview ...

Key	Summary
TP-12	Site preparations
SP-10	Move stuff to
SP-1	Relocate

3 items Open Info

Agile - ¿Cómo DevOps puede ayudar?



jira.teamsinspace.com

Teams in Space
Software project

- Backlog
- Board
- Reports
- Releases
- Components
- Issues
- Repository
- Add item
- Settings

Board

Quick Filters

TO DO 5	IN PROGRESS 5	CODE REVIEW 2	DONE 8
<p>Engage Jupiter Express for outer solar system travel</p> <p>SPACE TRAVEL PARTNERS</p> <p>✓ ↑ 5 TIS-25</p>	<p>Requesting available flights is now taking > 5 seconds</p> <p>SEESPACEEZ PLUS</p> <p>✓ ↑ 3 TIS-8</p>	<p>Register with the Mars Ministry of Revenue</p> <p>LOCAL MARS OFFICE</p> <p>✓ ↑ 3 TIS-11</p>	<p>Homepage footer uses an inline style- should use a class</p> <p>LARGE TEAM SUPPORT</p> <p>✓ ↑ TIS-68</p>
<p>Create 90 day plans for all departments in the Mars Office</p> <p>LOCAL MARS OFFICE</p> <p>✓ 9 TIS-12</p>	<p>Engage Saturn Shuttle Lines for group tours</p> <p>SPACE TRAVEL PARTNERS</p> <p>✓ ↑ 4 TIS-15</p>	<p>Draft network plan for Mars Office</p> <p>LOCAL MARS OFFICE</p> <p>✓ ↑ 3 TIS-15</p>	<p>Engage JetShuttle SpaceWays for travel</p> <p>SPACE TRAVEL PARTNERS</p> <p>✓ ↑ 5 TIS-23</p>
<p>Engage Saturn's Rings Resort as a preferred provider</p> <p>SPACE TRAVEL PARTNERS</p> <p>✓ ↑ 3 TIS-17</p>	<p>Establish a catering vendor to provide meal service</p> <p>LOCAL MARS OFFICE</p> <p>✓ ↑ 4 TIS-15</p>		<p>Engage Saturn Shuttle Lines for group tours</p> <p>SPACE TRAVEL PARTNERS</p> <p>✓ ↑ TIS-15</p>
<p>Enable Speedy SpaceCraft as the preferred</p>	<p>Engage Saturn Shuttle Lines for group tours</p>		<p>Establish a catering vendor to provide meal service</p> <p>LOCAL MARS OFFICE</p>

<http://agilemanifesto.org/iso/es/principles.html>

Principios del Manifiesto Ágil

Seguimos estos principios:

Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.

Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.

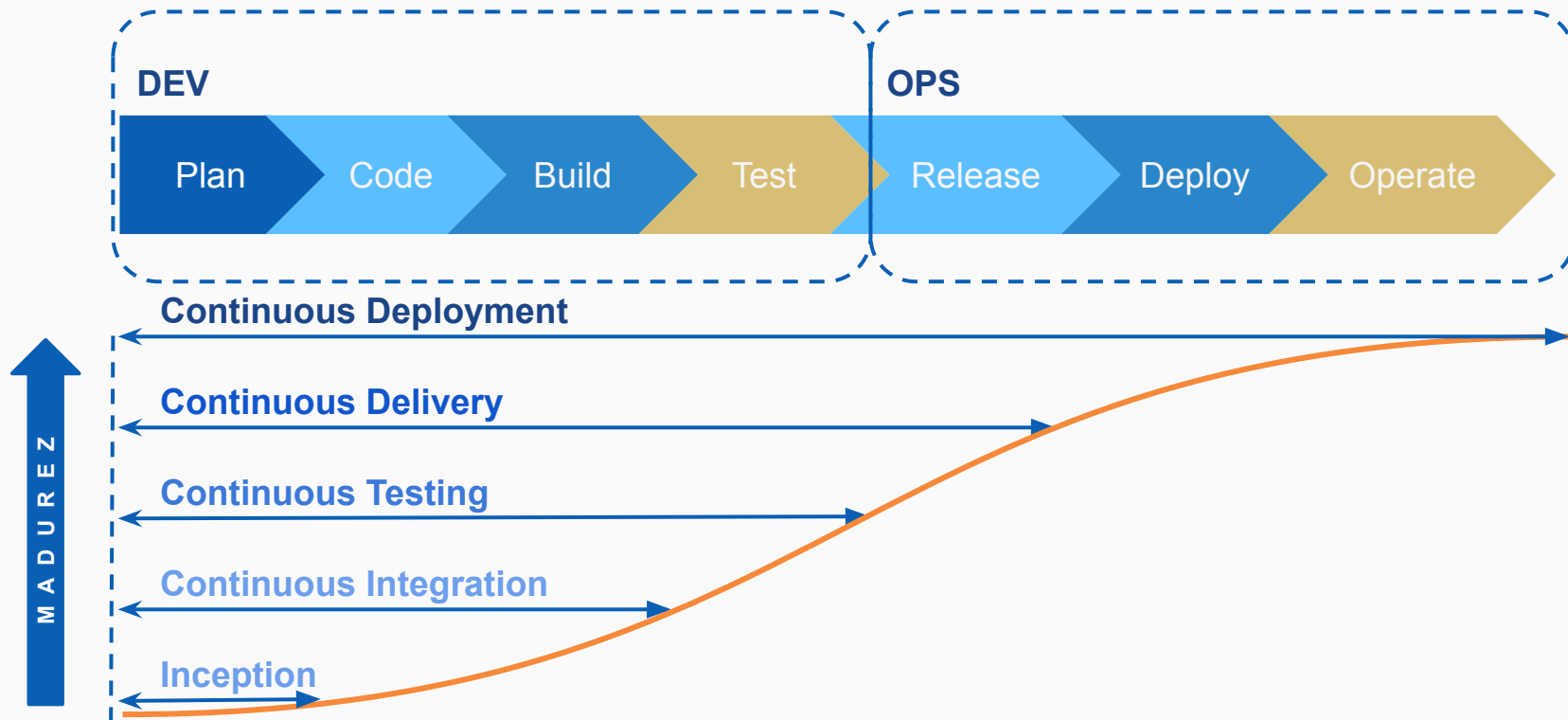
Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.

Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.

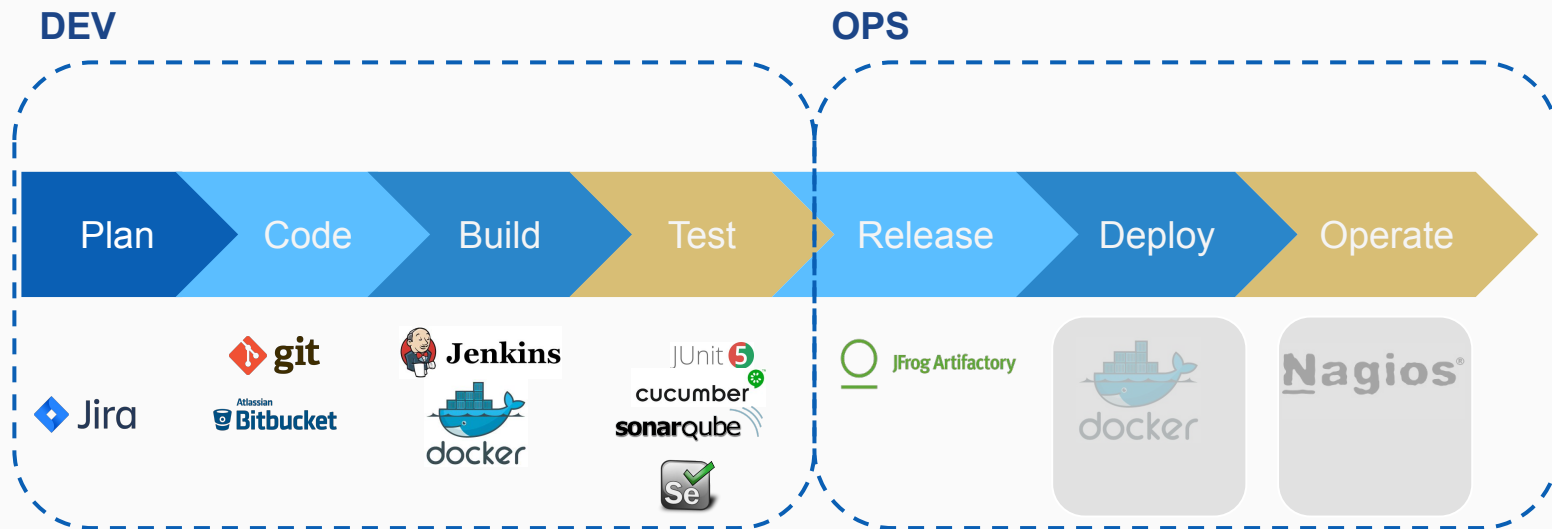


Con DevOps, NO todo es deployment, Sí se puede tener el tracking desde la planificación hasta el despliegue con las herramientas correctas





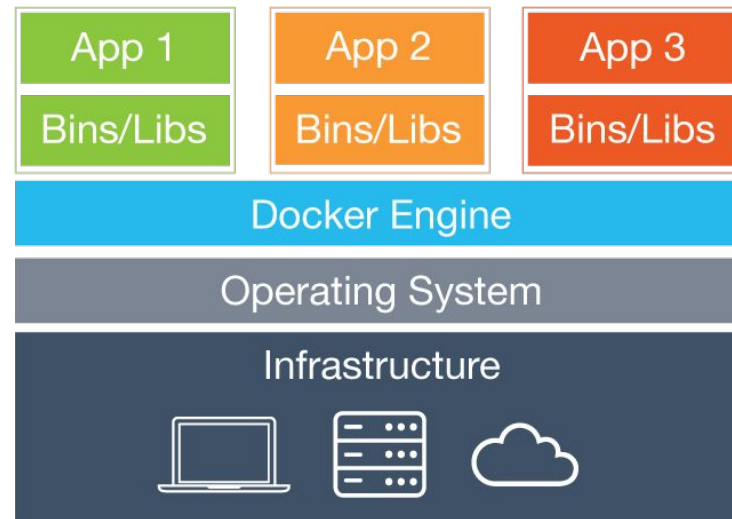
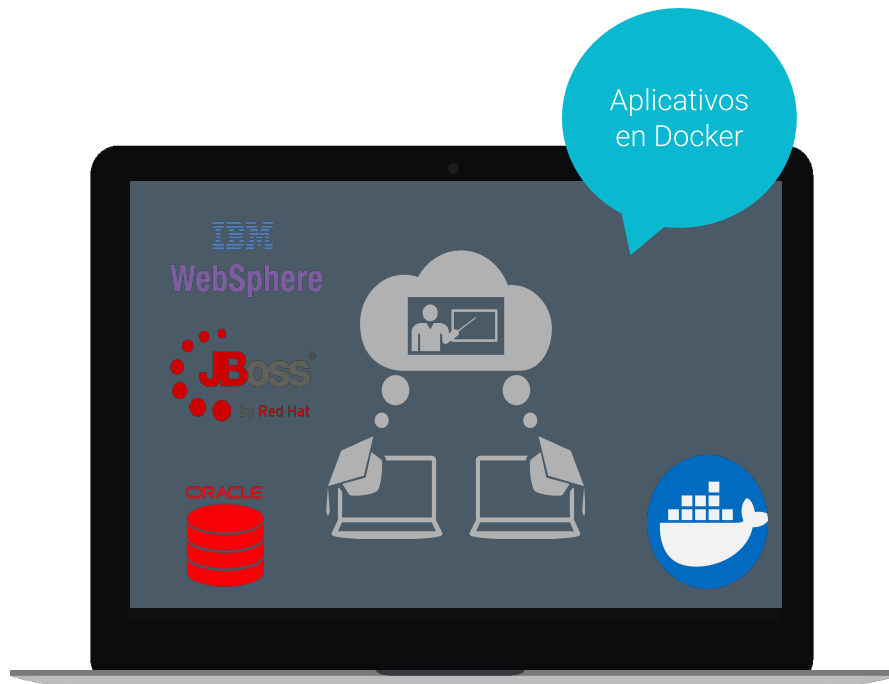
Algunas Herramientas... de las tantas que hay



Docker



¿Qué es Docker?



Docker es una herramienta que empaqueta o “embala” aplicaciones y sus dependencias en imágenes. Éstas son inmutables, ligeras y portables. Y permite ejecutar “N” veces estas imágenes en contenedores.

¿Qué versiones de Docker

tenemos?



INSTALACIÓN DE DOCKER



Windows 8.1

**Windows 10
(Home) 8,7 y
otros**

[Docker Toolbox](#)
[gitbash \(opcional\)](#)



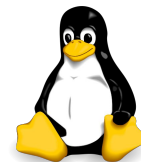
**Windows 10 (Pro o
Enterprise)**

[Docker for windows](#)
[Docker Toolbox](#)
[\(opcional\)](#)
[gitbash\(opcional\)](#)



Mac

[Mac OS >= Yosemite :](#)
[Docker for mac](#)
[Docker toolbox](#)
[\(opcional\)](#)



Linux

docker: `curl -sSL`
`https://get.docker.com`
`/ | sh`
[Docker compose](#)
[Docker machine](#)



Verificando instalación de docker

1

\$ docker version

Se debe mostrar información de 2 puntos importantes:

- a.- El demonio docker
- b.- El cliente docker

2

\$ docker info

Se muestra mucha información del uso de docker, como por ejemplo: memoria, red, etc

3

\$ docker run hello-world

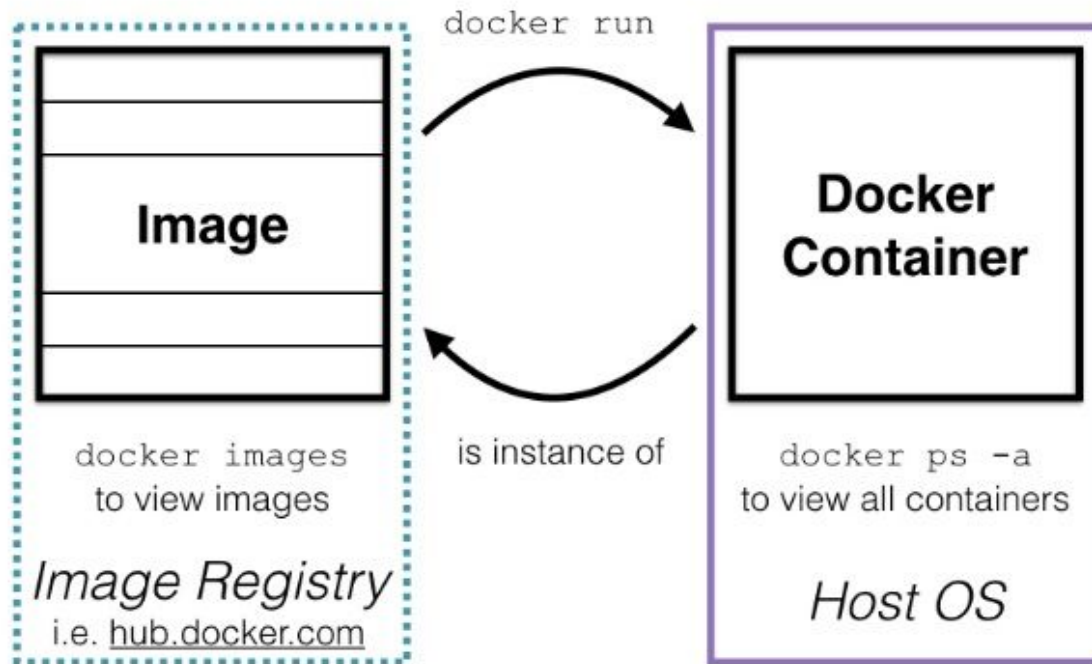
Se crea un contenedor y este es el famoso "hola mundo".

```
$ docker version
Client:
 Version:      18.03.0-ce
 API version:  1.37
 Go version:   go1.9.4
 Git commit:   0520e24302
 Built: Fri Mar 23 08:31:36 2018
 OS/Arch:     windows/amd64
 Experimental: false
 Orchestrator: swarm

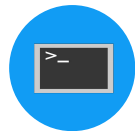
Server:
 Engine:
  Version:      18.05.0-ce
  API version:  1.37 (minimum version 1.12)
  Go version:   go1.10.1
  Git commit:   f150324
  Built:       Wed May  9 22:20:42 2018
  OS/Arch:     linux/amd64
  Experimental: false
```



Imagen y Contenedor



Creando nuestro 1er contenedor



Terminal

```
$ docker run -d -p 80:80 --name servidor_nginx nginx
```

```
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
27833a3ba0a5: Pulling fs layer
e83729dd399a: Pulling fs layer
ebc6a67df66d: Pulling fs layer
ebc6a67df66d: Verifying Checksum
ebc6a67df66d: Download complete
e83729dd399a: Verifying Checksum
e83729dd399a: Download complete
27833a3ba0a5: Verifying Checksum
27833a3ba0a5: Download complete
27833a3ba0a5: Pull complete
e83729dd399a: Pull complete
ebc6a67df66d: Pull complete
Digest: sha256:c8a861b8a1eeef6d48955a6c6d5dff8e2580f13ff4d0f549e082e7c82a8617a2
Status: Downloaded newer image for nginx:latest
dacd4807bece8339fed41bdf9dc0779a1ef01301890b09e2dd6d5d4516ae4731
```

Se crea un **contenedor en segundo plano (-d)** **y se expone en el puerto 80 del host**. El contenedor tiene por nombre `servidor_nginx` y usa la **imagen nginx**

Si la imagen no está descargada en el host, docker busca en el registry público por defecto (docker hub) o en un registry privado

El contenedor creado tiene un **ID**

Resultado del 1er contenedor



Ingresar en el browser la siguiente URL: <http://localhost>

** Si instalaste docker toolbox, ingresa esta url: <http://192.168.99.100>

** Con docker toolbox no se debe usar la palabra localhost



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

```
$ docker run -d -p 80:80 --name servidor_nginx nginx
```

Sin -p (--publish), no podemos ingresar a la web de nginx,
--publish publica un puerto interno de un contenedor hacia un
puerto externo del host

Comandos básicos...

Estructura de los [comandos docker](#):

- Antiguo: docker <command> (options)
- Nuevo: docker <command> <subcommand> (options)

\$ docker images (sirve para visualizar las imágenes docker)

\$ docker ps -a (sirve para mostrar los contenedores incluidos los detenidos)

\$ docker run -it --rm -p 8888:8080 tomcat:8.0 (sirve para crear un contenedor en base a una imagen)

\$ docker rm -f tomcat_server (sirve para eliminar contenedores a la fuerza -f)

\$ docker rm -f \$(docker ps -aq) (sirve para eliminar todos los contenedores, incluido los detenidos)

\$ docker rmi -f mysql (sirve para eliminar una imagen a la fuerza -f)



Gestión de múltiples contenedores...

```
$ docker run -d -p 8000:8080 --name tomcat_server tomcat:8.0-alpine

$ docker run -d -p 9000:8080 -p 9990:9990 --name jboss_server andreptb/wildfly:8.2.0.Final.jdk8-alpine
/opt/jboss/wildfly/bin/domain.sh -b 0.0.0.0 -bmanagement 0.0.0.0
$ docker exec -it jboss_server /opt/jboss/wildfly/bin/add-user.sh admin Admin#70365 --silent

$ docker run -d -p 9043:9043 -p 9443:9443 --name was_server ibmcom/websphere-traditional:latest
$ docker exec -it was_server cat /tmp/PASSWORD

$ docker logs -f jboss_server

$ docker exec -it jboss_server bash

$ docker rm -f tomcat_server
$ docker rm -f $(docker ps -aq)

$ docker rmi -f mysql
```



Gestión de múltiples contenedores...

```
$ docker pull mysql:latest

$ docker run -d -p 3333:3306 -e MYSQL_ROOT_PASSWORD=12345678 --name mysql_server mysql:5.7.25
--default-authentication-plugin=mysql_native_password

$ docker cp ~/Desktop/fuentes_sesion_05/apps/mysql_server/scripts/DDL.sql mysql_server:/tmp

$ docker exec -it mysql_server bash
    $root@a123e:/# mysql -u root -p
    $mysql> source /tmp/DDL.sql
    $mysql> exit
    $root@a123e:/# exit

$ docker stop mysql_server

$ docker rm mysql_server
```





¿Dudas?

devops@mitocodenetwork.com

