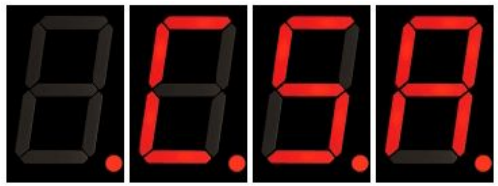
UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
School of Professional & Executive Development

AGILE IT MANAGEMENT
SCRUM, **DEVOPS**, LEAN IT

prácticas de devops

Carles San Agustín

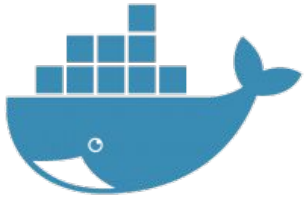
www.carlessanagustin.com



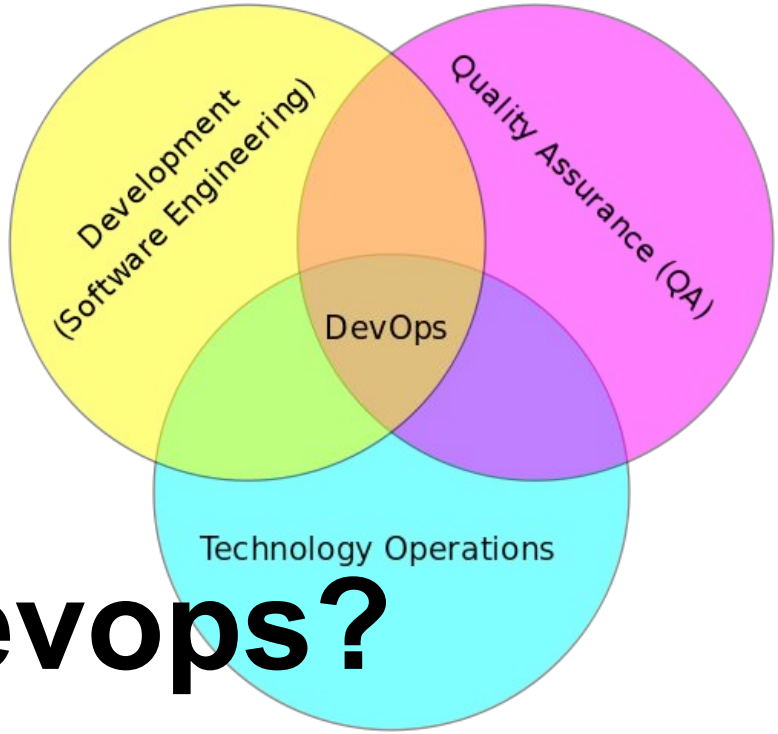
@carlesanagustin

linkedin.carlessanagustin.com

índice: programario



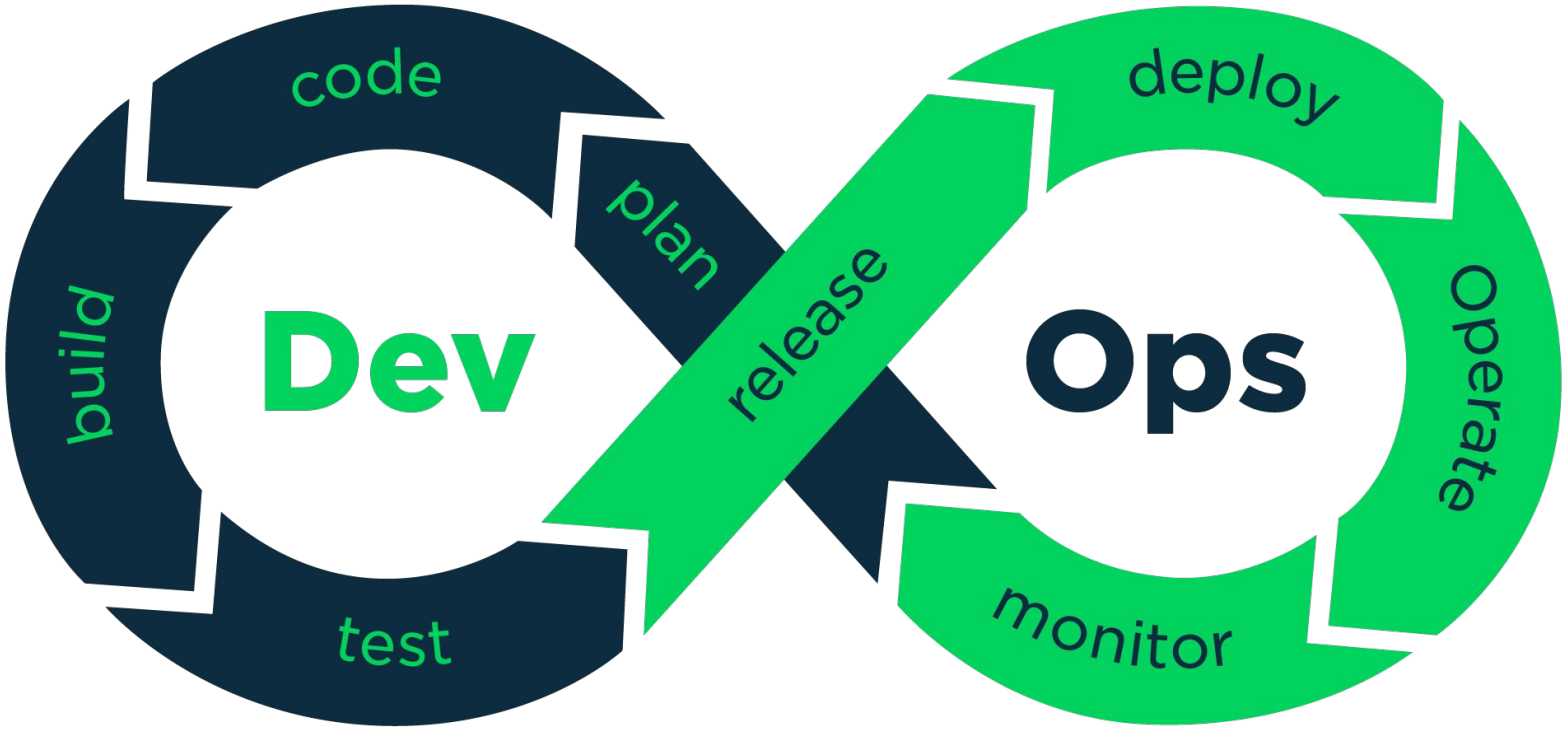
1. ¿qué es devops?
2. desarrollo
   - git (práctica) + vagrant (práctica)
3. despliegue
   - ansible (práctica) + nagios (práctica)
4. entornos
   - docker (práctica)



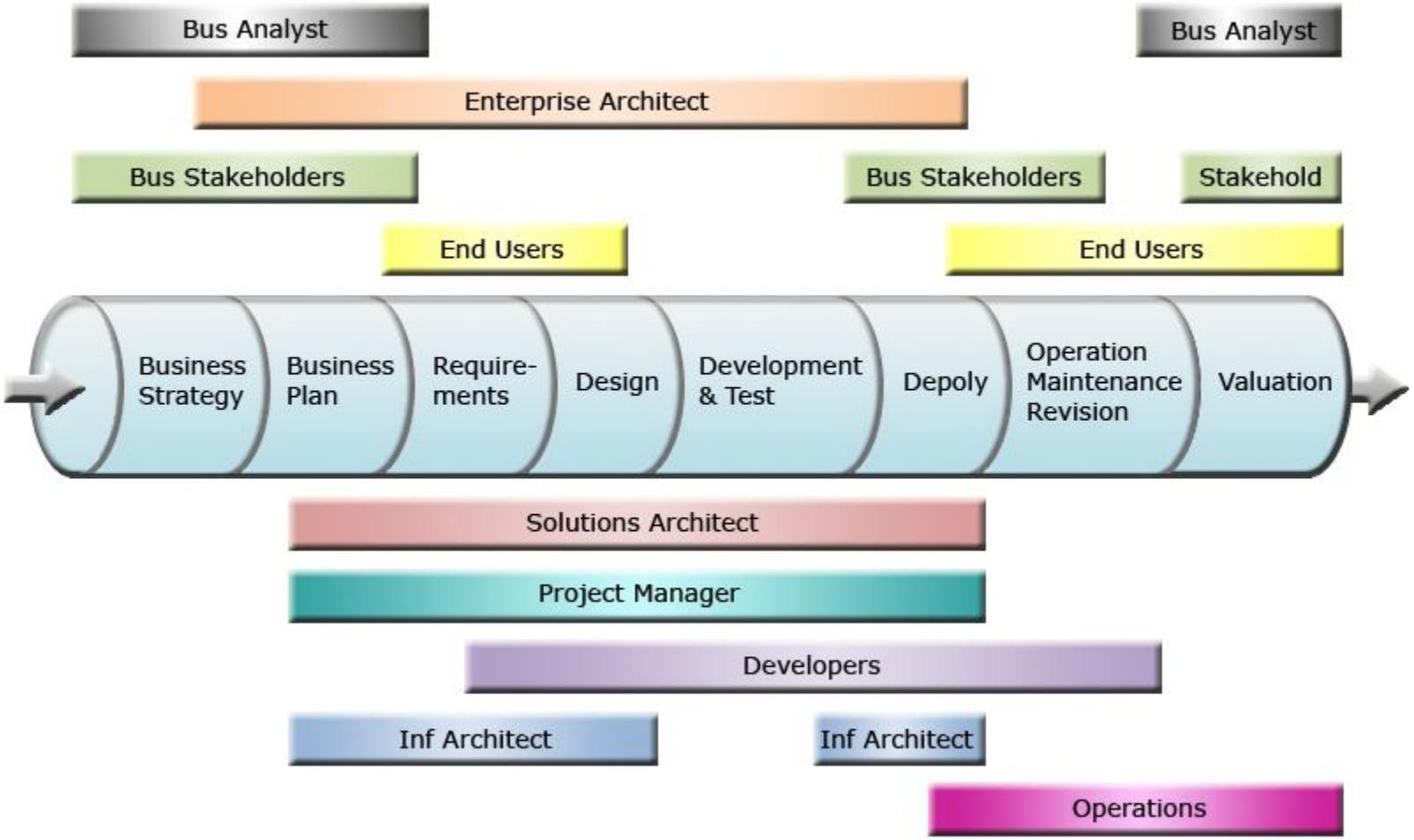
¿qué es devops?

desarrollo y operaciones

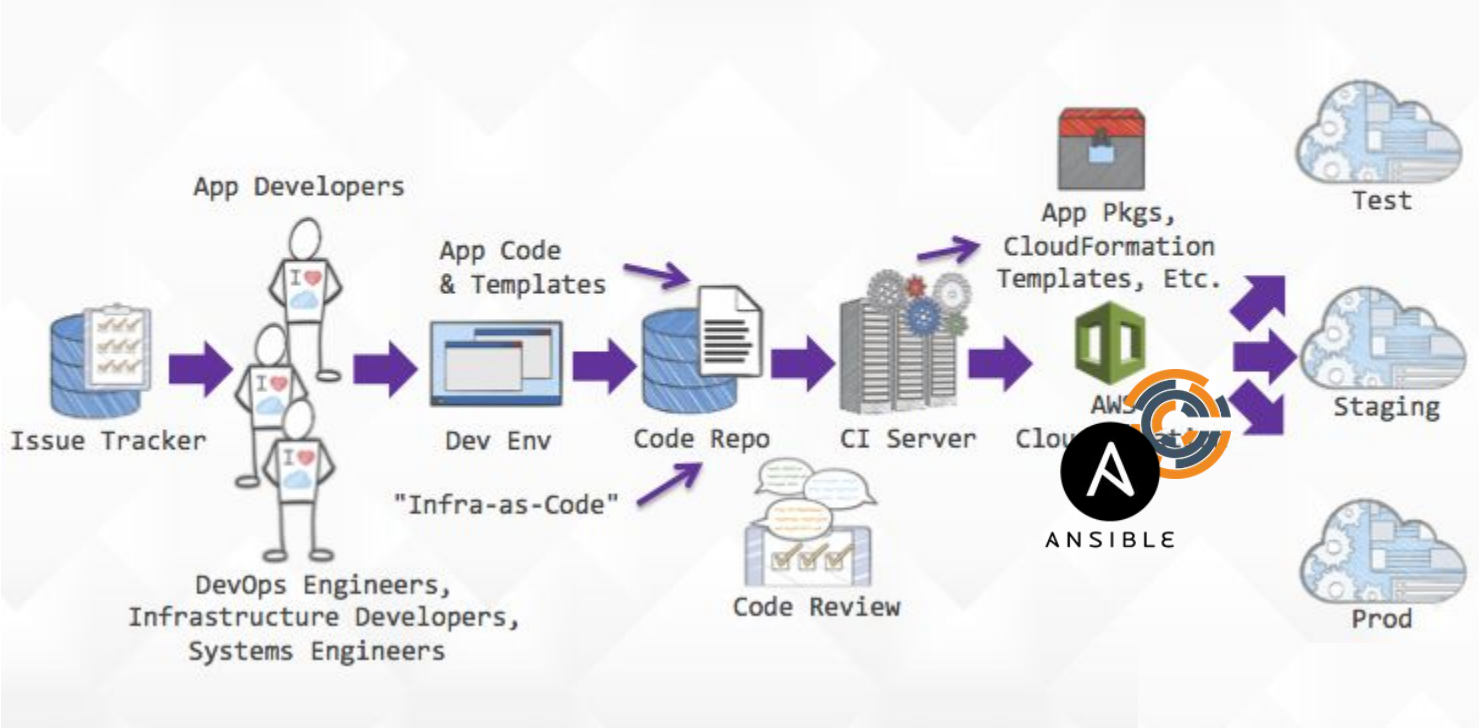
¿qué es devops? - bucle de publicación y realimentación



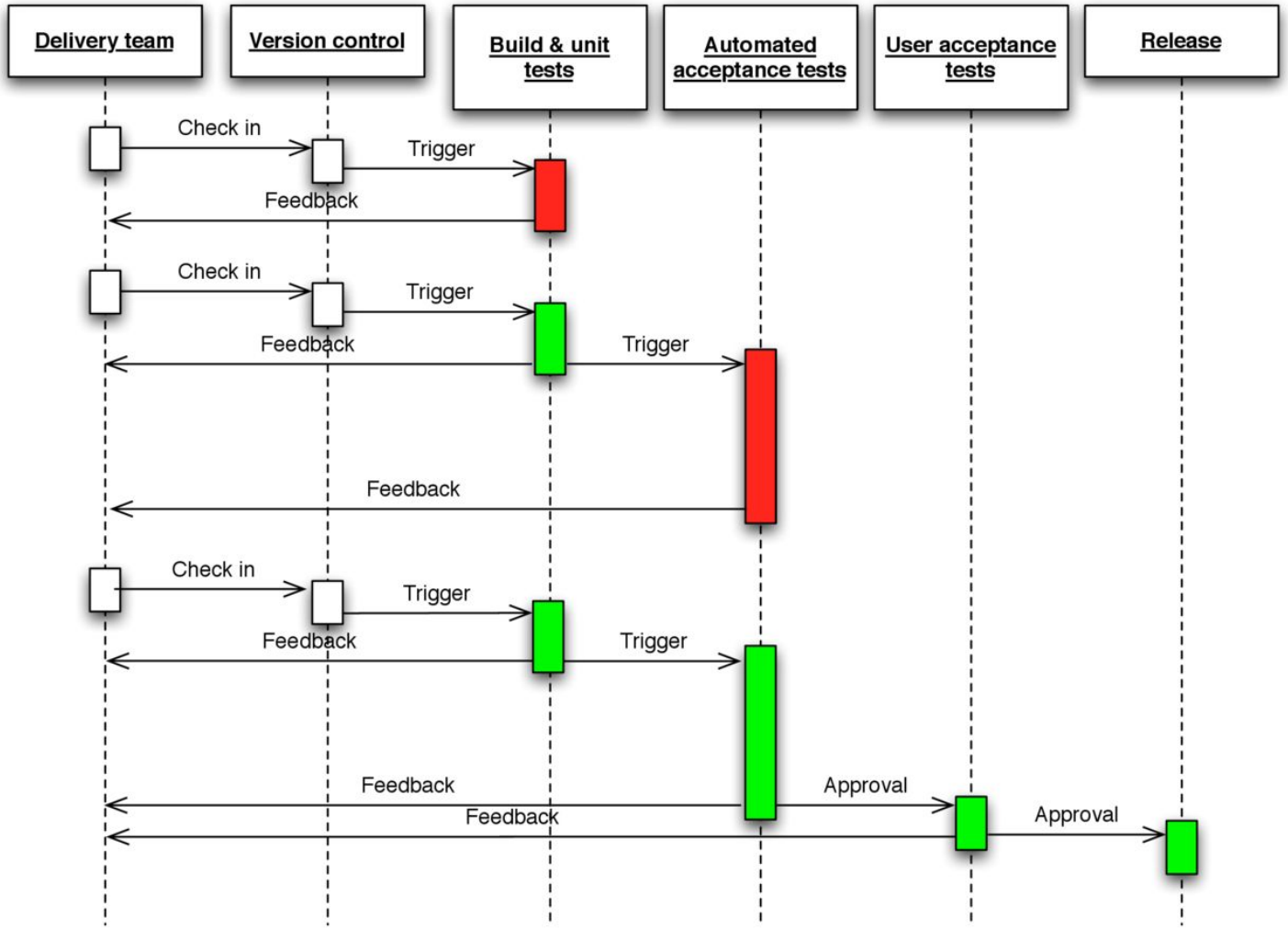
¿qué es devops? - ciclo de vida de una solución



¿qué es devops? - infraestructura como código (IaC)

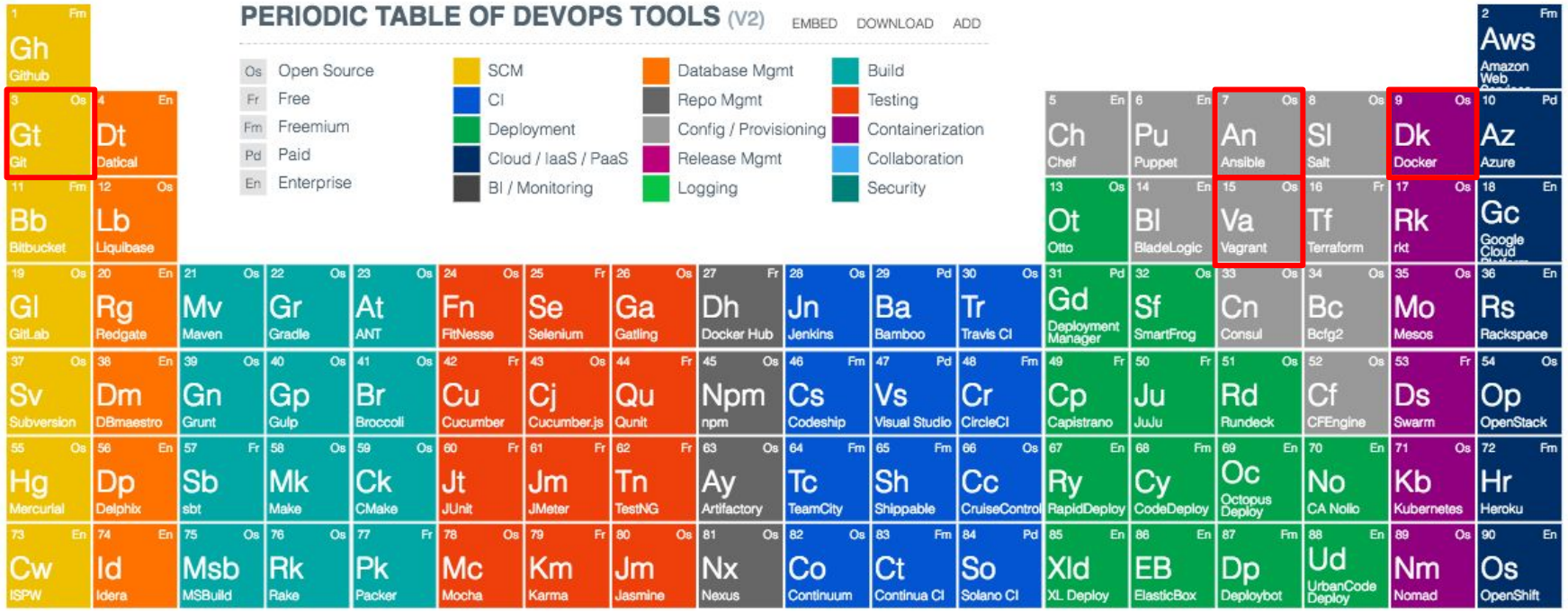


¿qué es devops? - tubería de despliegue (CI)



fuente: jez humble

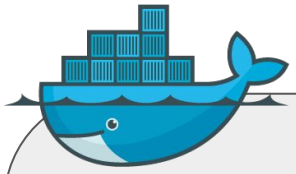
¿qué es devops? - herramientas



XebiaLabs
Enterprise DevOps

Follow @xebialabs
Publication Guidelines

entorno de prácticas



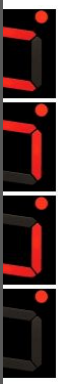
ZIPI
(192.168.56.10)

DOCKER engine
NAGIOS node

ZAPE
(192.168.56.11)

ANSIBLE server
NAGIOS server

http://localhost:8082/nagios3/



herramientas para las prácticas

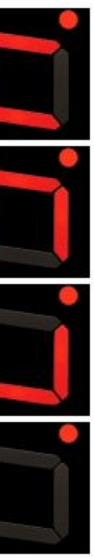
- virtualbox: https://www.virtualbox.org/
- vagrant: http://www.vagrantup.com/
- putty: http://www.chiark.greenend.org.uk/
- sublime text: http://www.sublimetext.com/
- git: http://git-scm.com/
- sourcetree: http://www.sourcetreeapp.com/

desarrollo

día 1

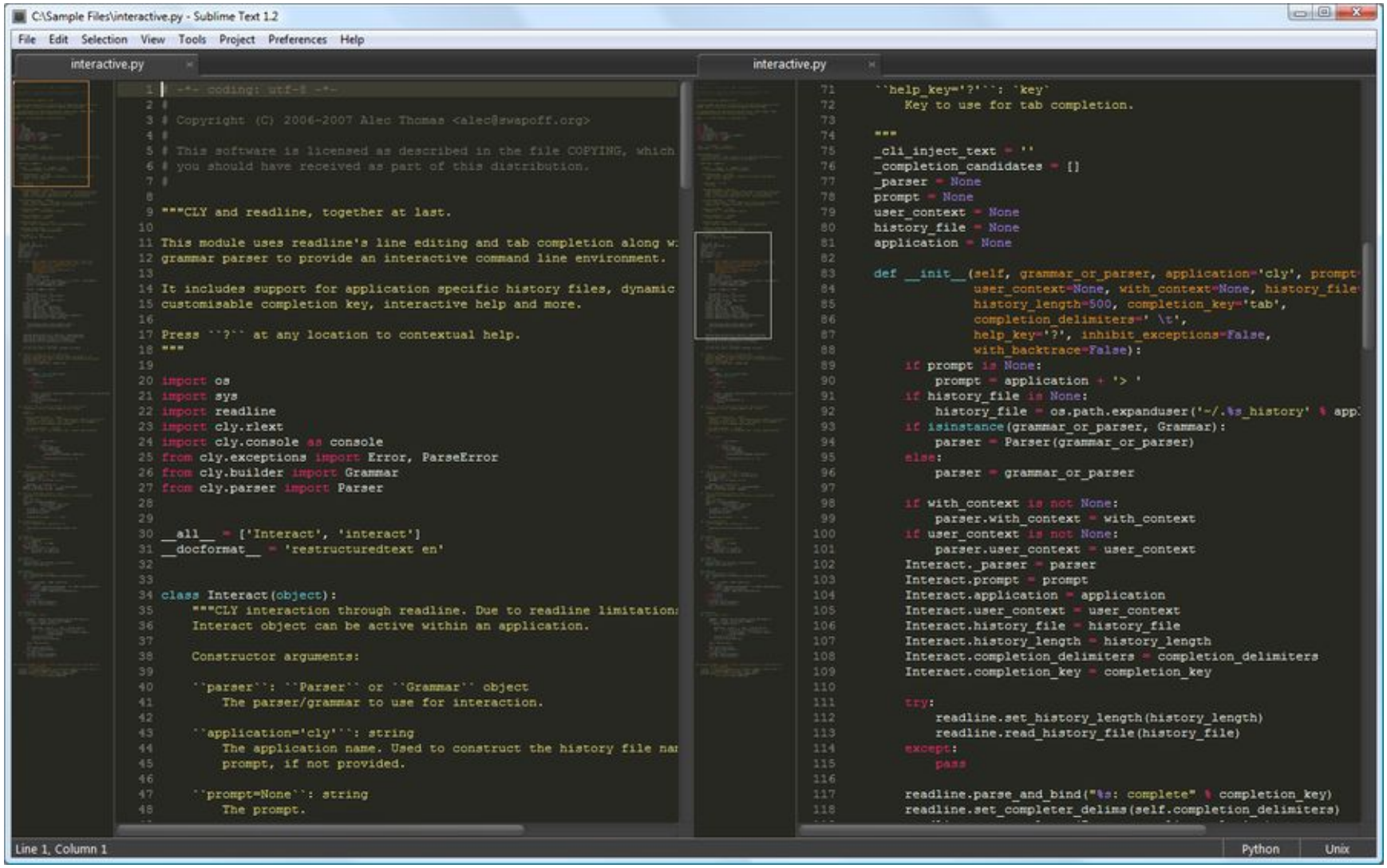
desarrollo: índice

1. editor
2. entorno virtualizado
3. control de versiones



editores: herramientas

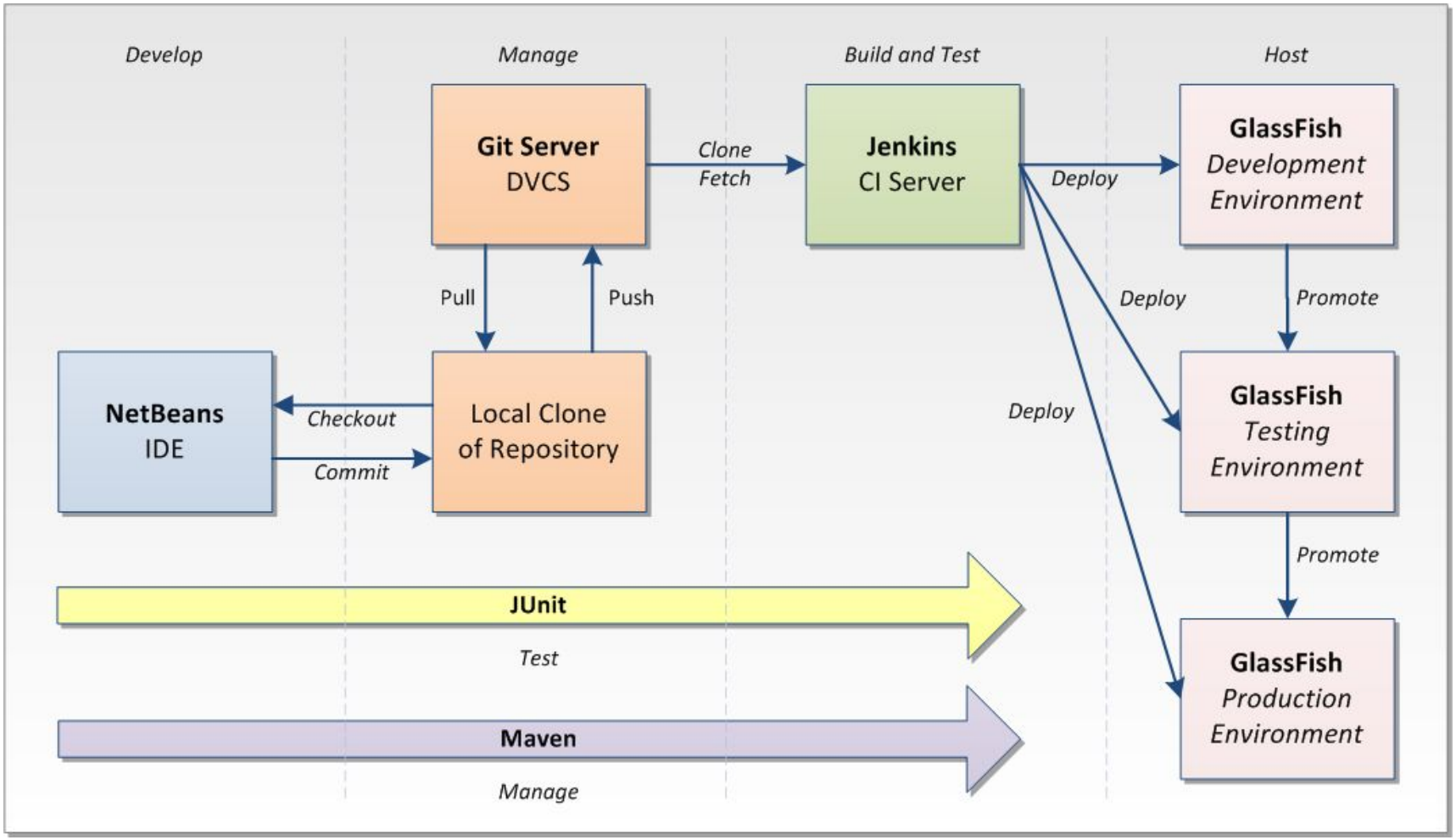
editores - sublime text



ejemplos reales

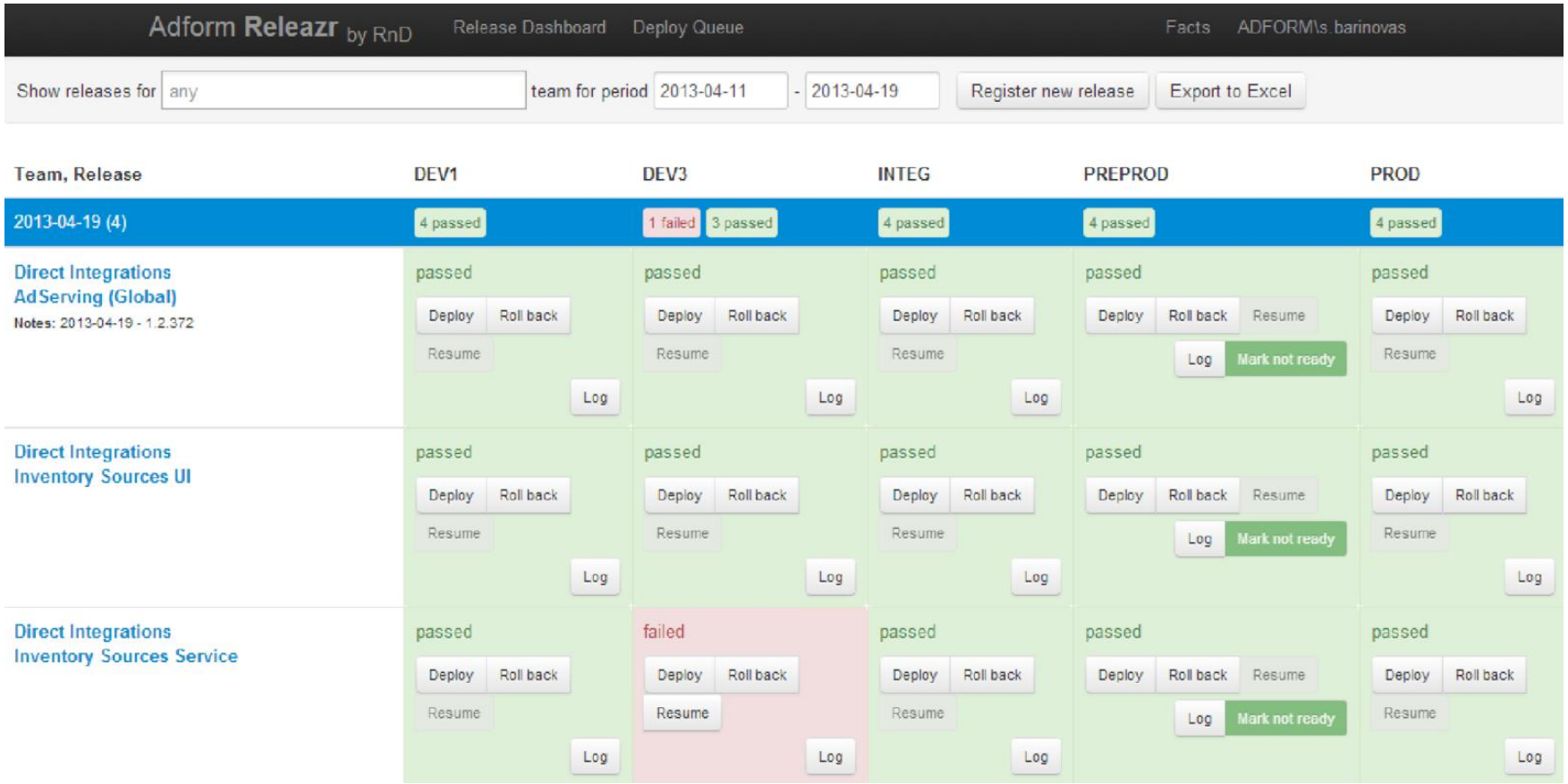
día 1

ejemplo 1.1: Java CI



fuente: http://goo.gl/3sjkL2

ejemplo 1.2: Sergejus Barinova on deployment @ AdForm

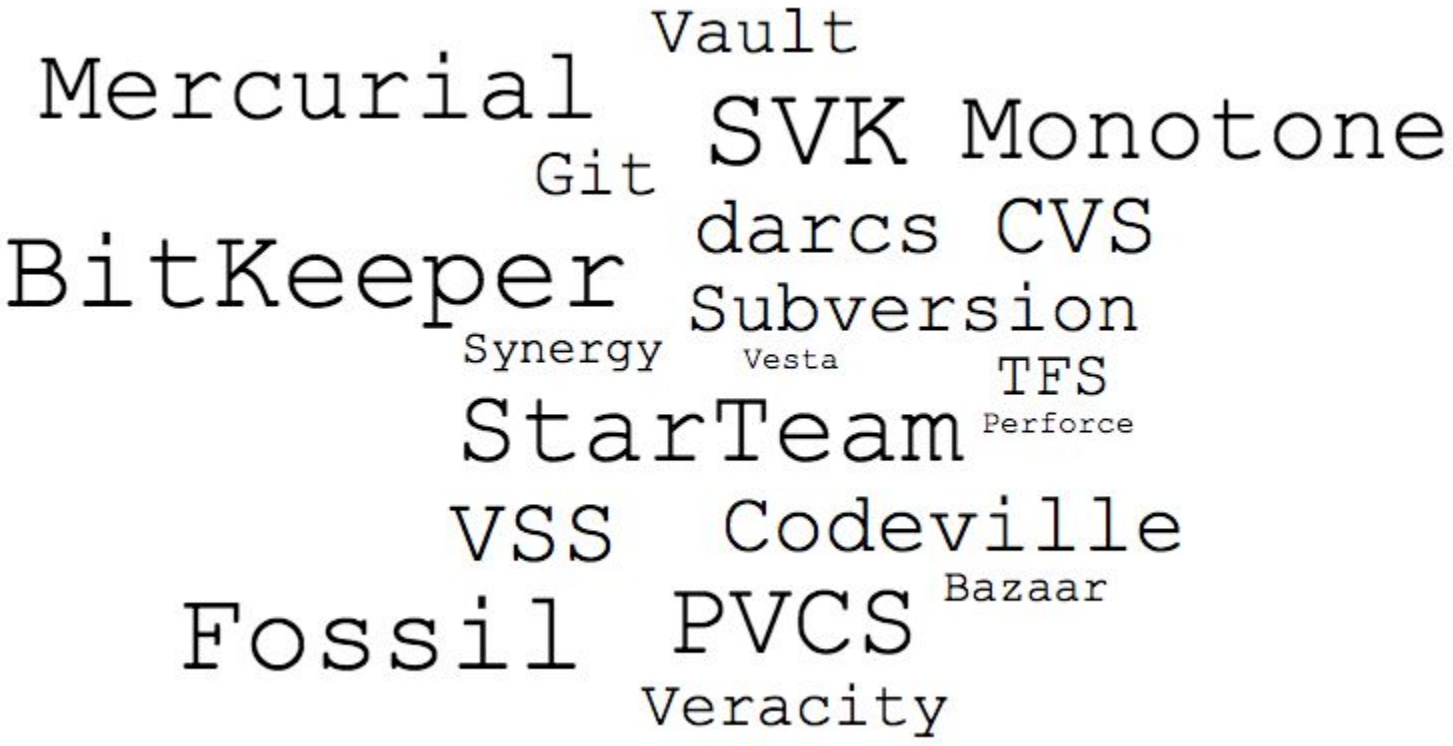


fuente: http://www.infoq.com/presentations/continuous-delivery-happiness

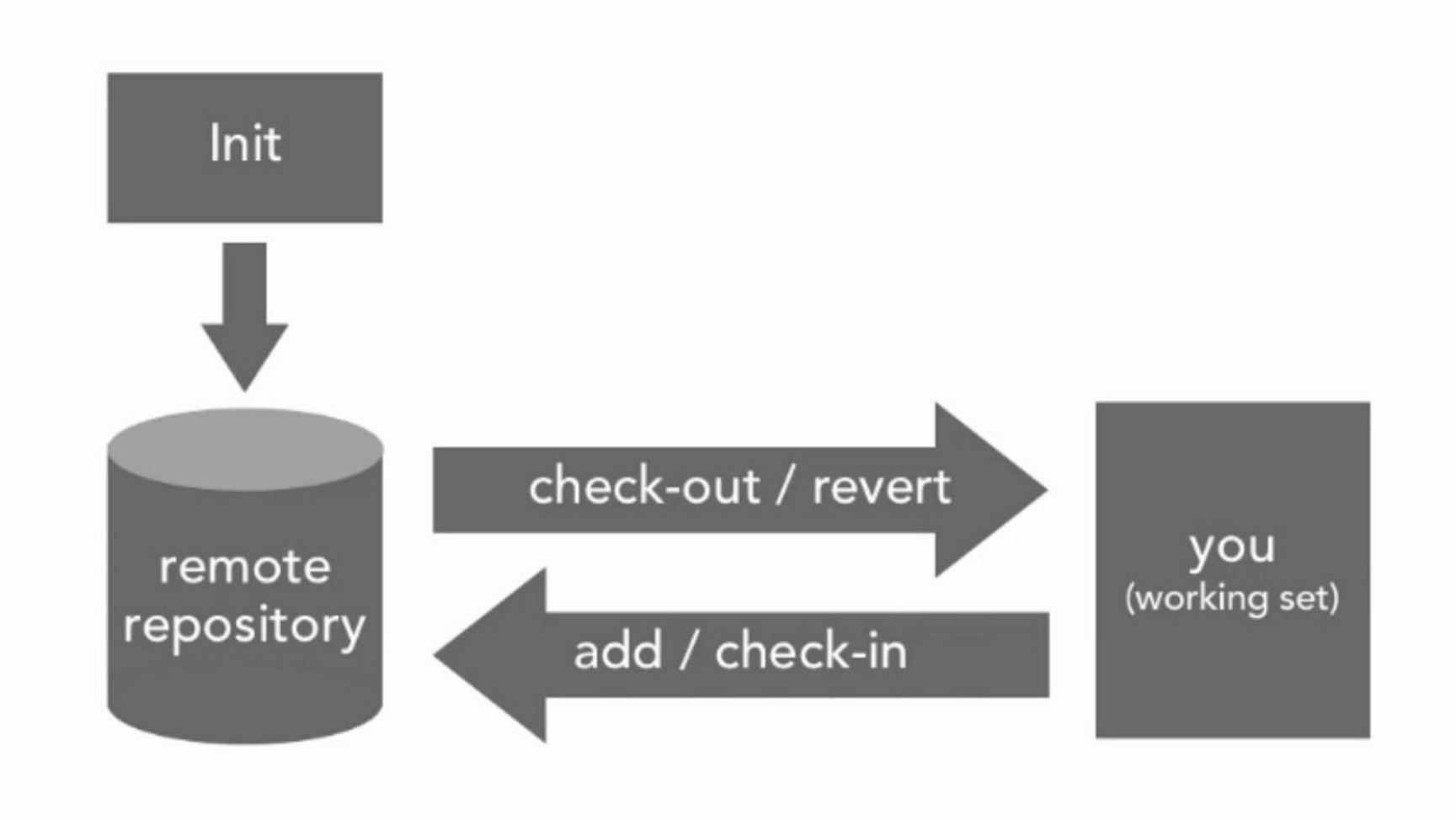
desarrollo: git

control de versiones

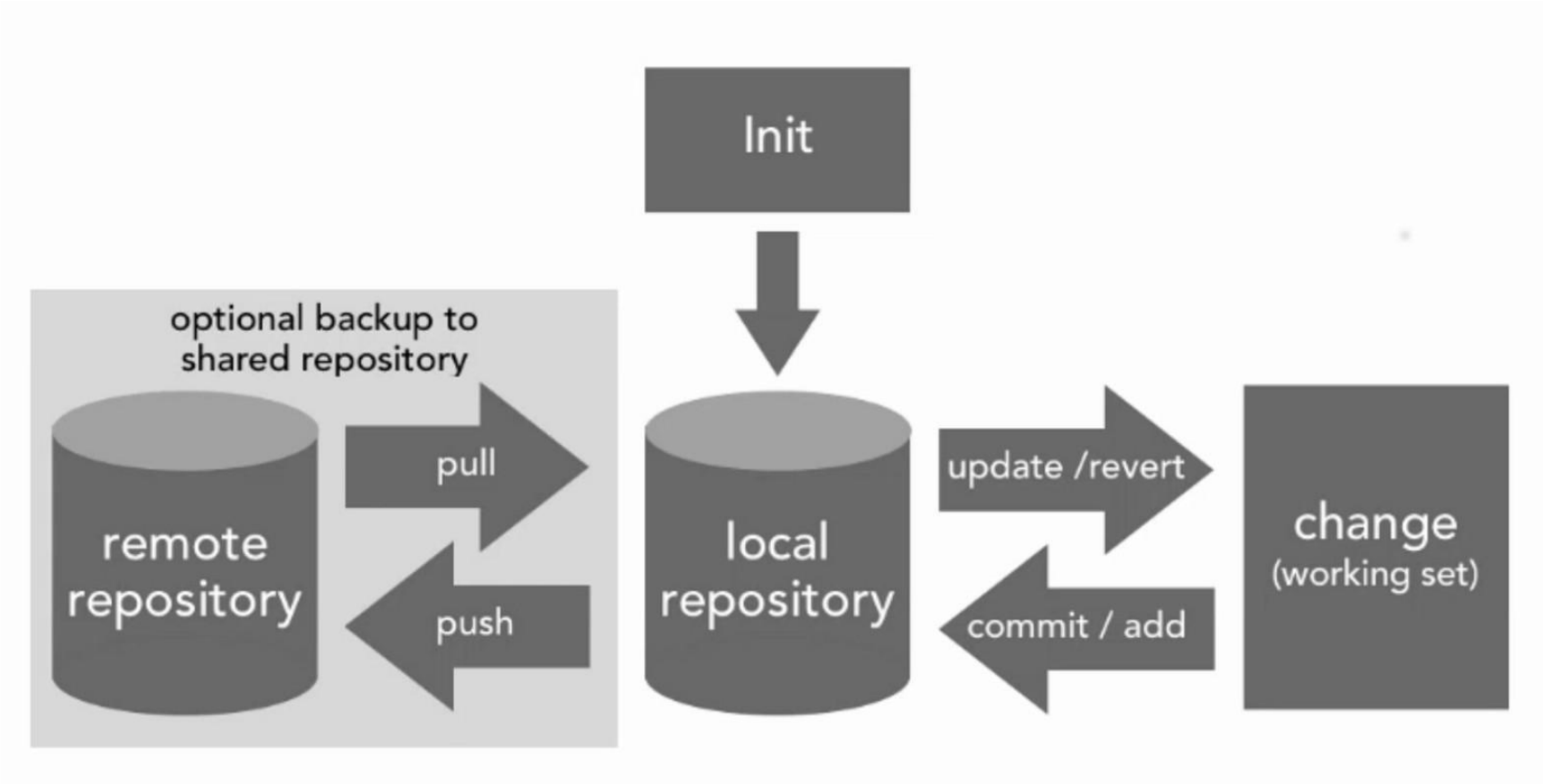
control de versiones: herramientas



control de versiones: centralizado



control de versiones: distribuido



control de versiones: centralizado contra distribuido

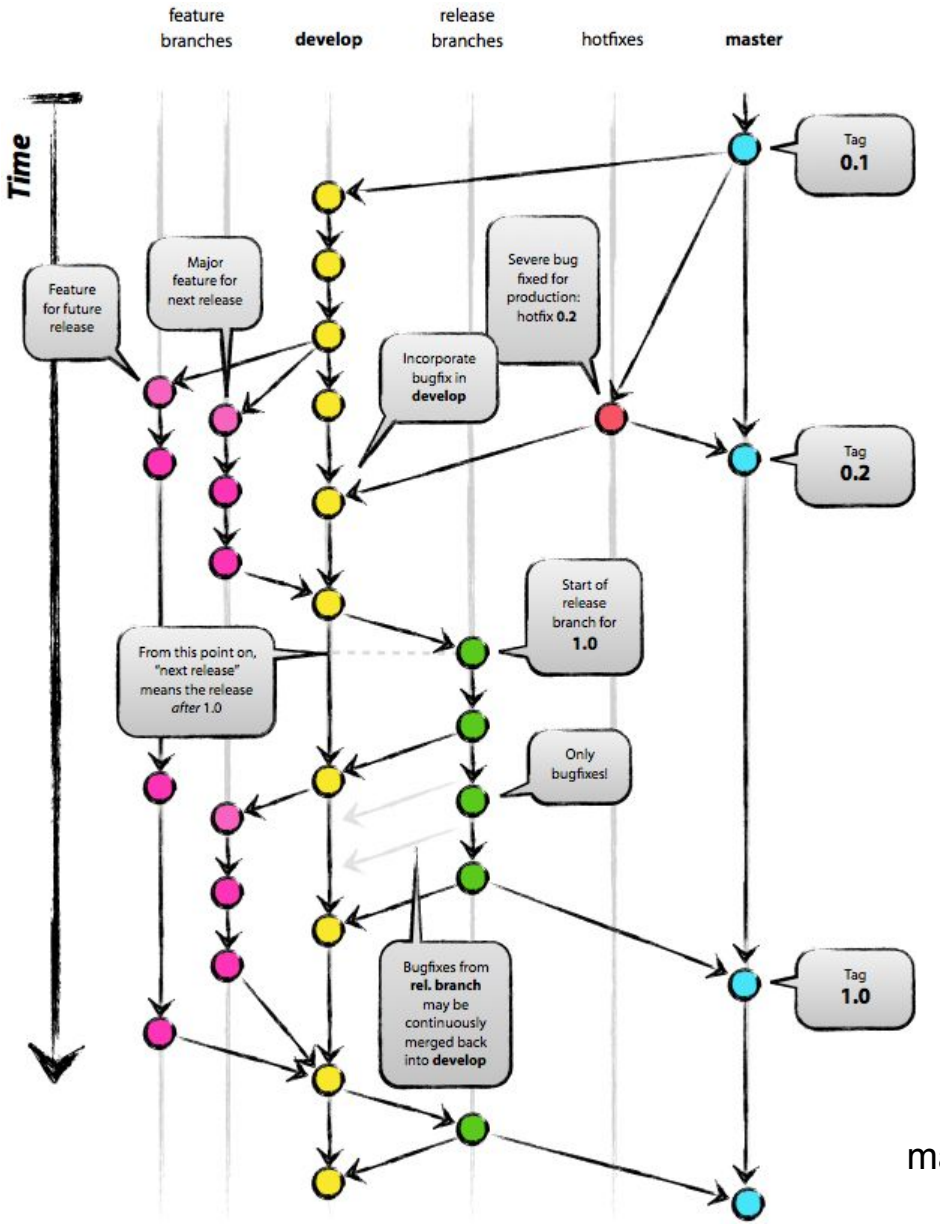
Centralized	
CVS	1986
Perforce	1995
Subversion	2000
MS Team Foundation Server	2010

Distributed	
Git	2005
Mercurial	2005

control de versiones: beneficios

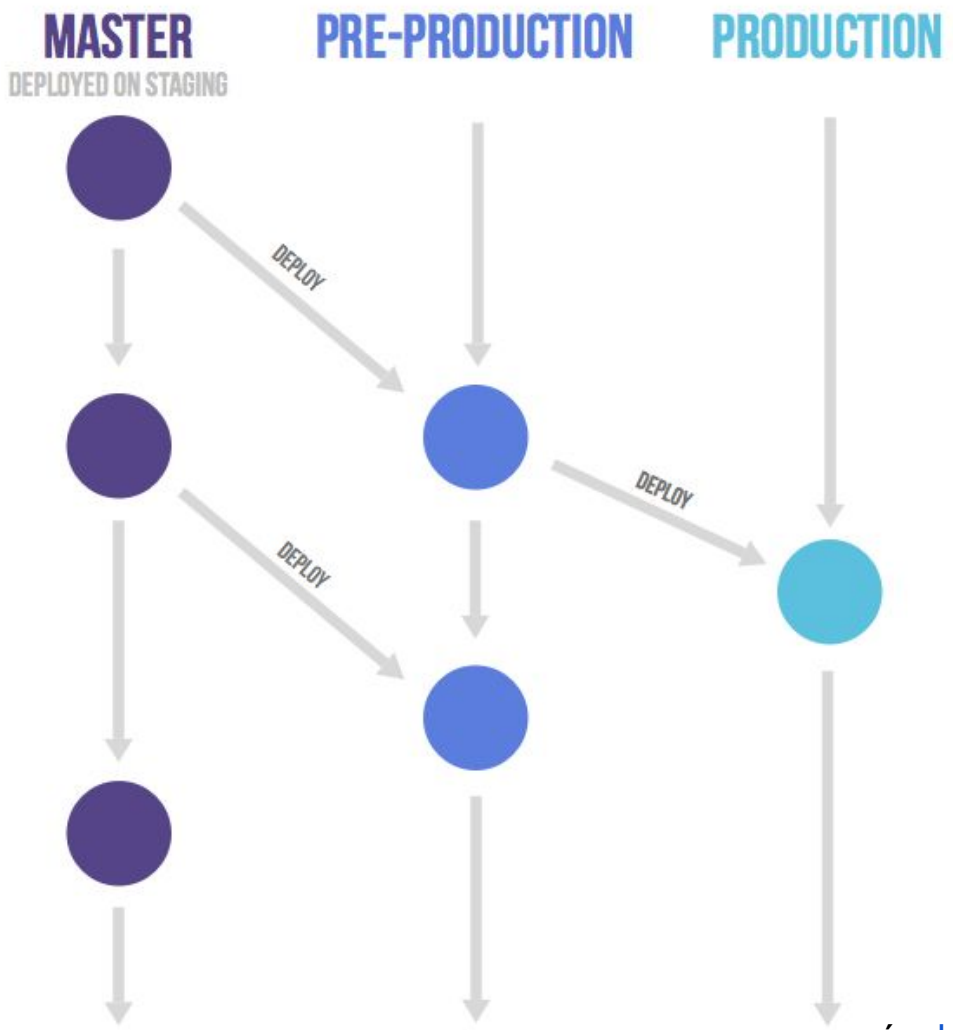
- Sincronización
- Responsabilidad
- Detección de conflictos
- Seguimiento de cambios
- Deshacer a la versión X
- Etiquetado de cambios
- *Copias de seguridad*

ramas: git flow



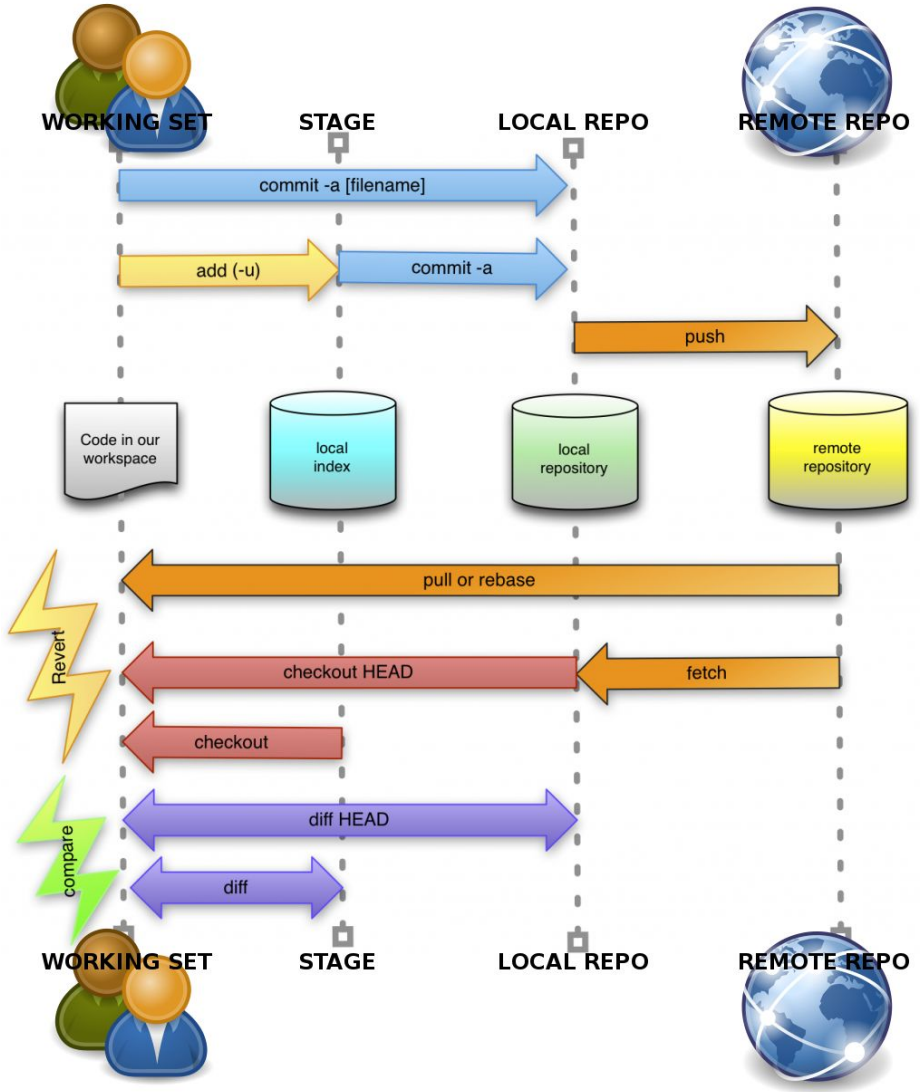
más: http://goo.gl/Tf1vn

ramas: gitlab flow



más: https://goo.gl/Z0w1nQ

desarrollo: git - comandos de transporte de datos



desarrollo: git - más información

Git - http://git-scm.com

http://gitref.org/

https://www.atlassian.com/git/

http://rogerdudler.github.io/git-guide/

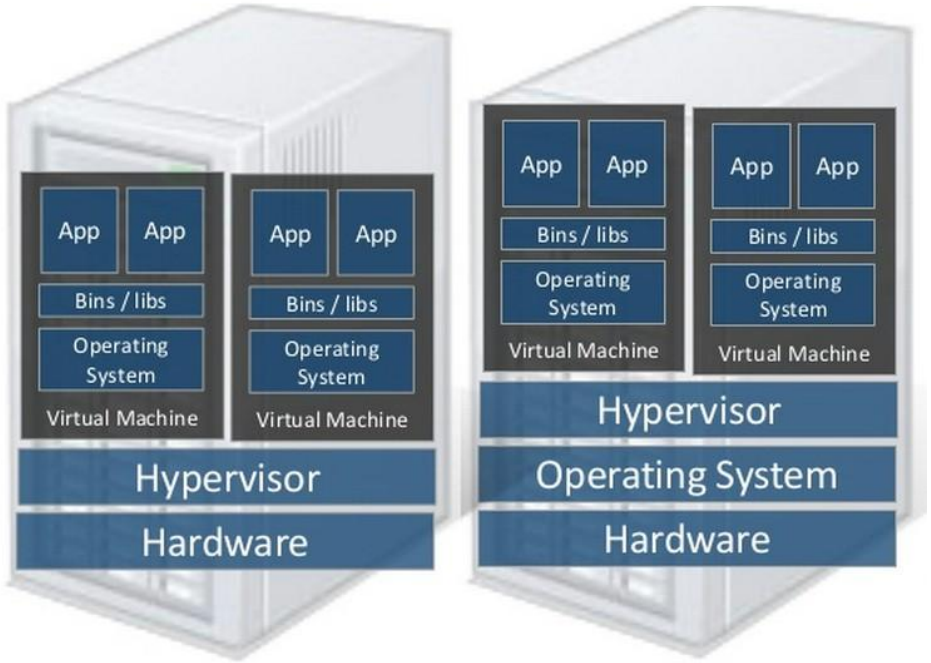
desarrollo:
git - práctica

http://try.github.io/

desarrollo: vagrant

entorno de desarrollo virtualizado

entorno virtualizado: ¿que es?



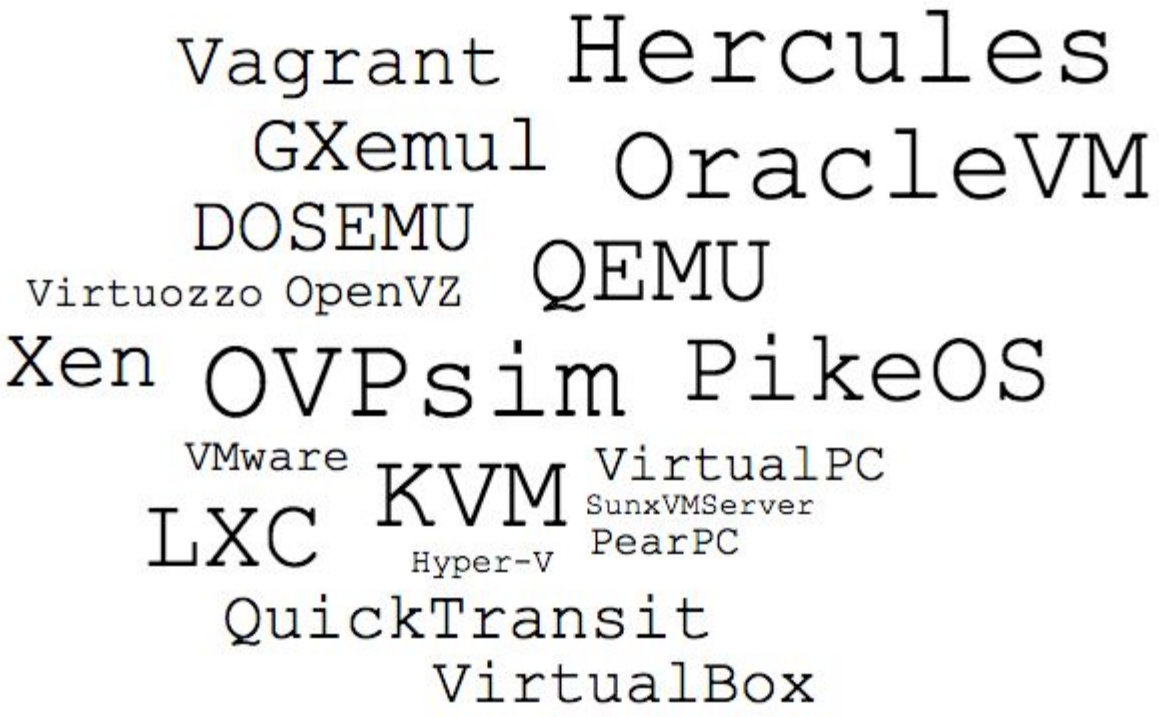
Type 1 Hypervisor

Oracle VM Server, Citrix XenServer, VMware ESX/ESXi, Microsoft Hyper-V, ...

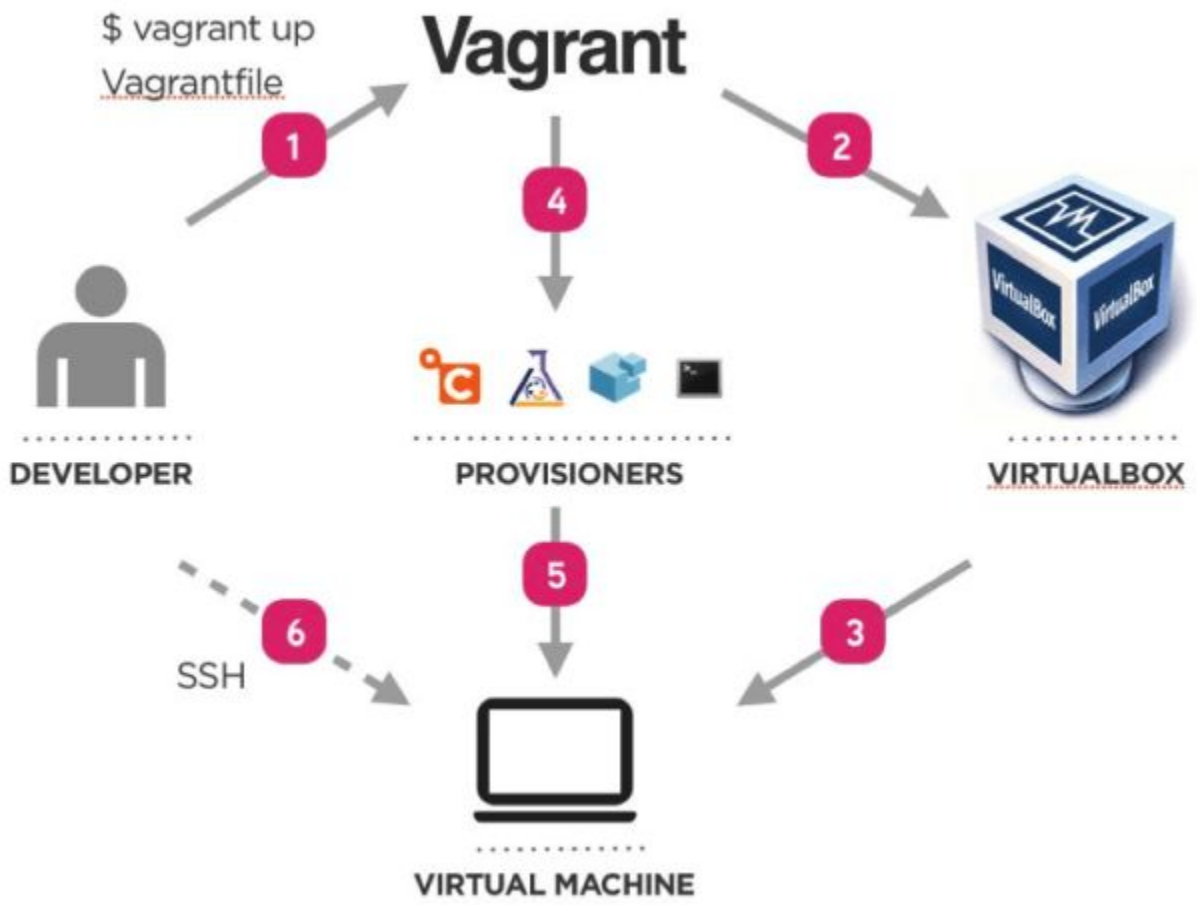
Type 2 Hypervisor

VMware Workstation, VMware Player, VirtualBox, Parallels, KVM, QEMU, ...

entorno virtualizado: herramientas



desarrollo: vagrant - ¿que es?

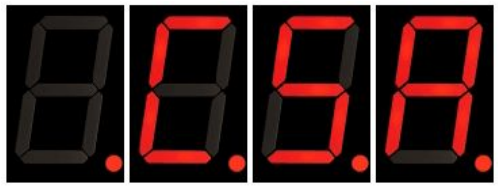


desarrollo:
vagrant - práctica

https://github.com/carlessanagustin/AITM

Carles San Agustín

www.carlessanagustin.com



@carlesanagustin

linkedin.carlessanagustin.com

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
School of Professional & Executive Development

AGILE IT MANAGEMENT
SCRUM, **DEVOPS**, LEAN IT

prácticas de devops

despliegue

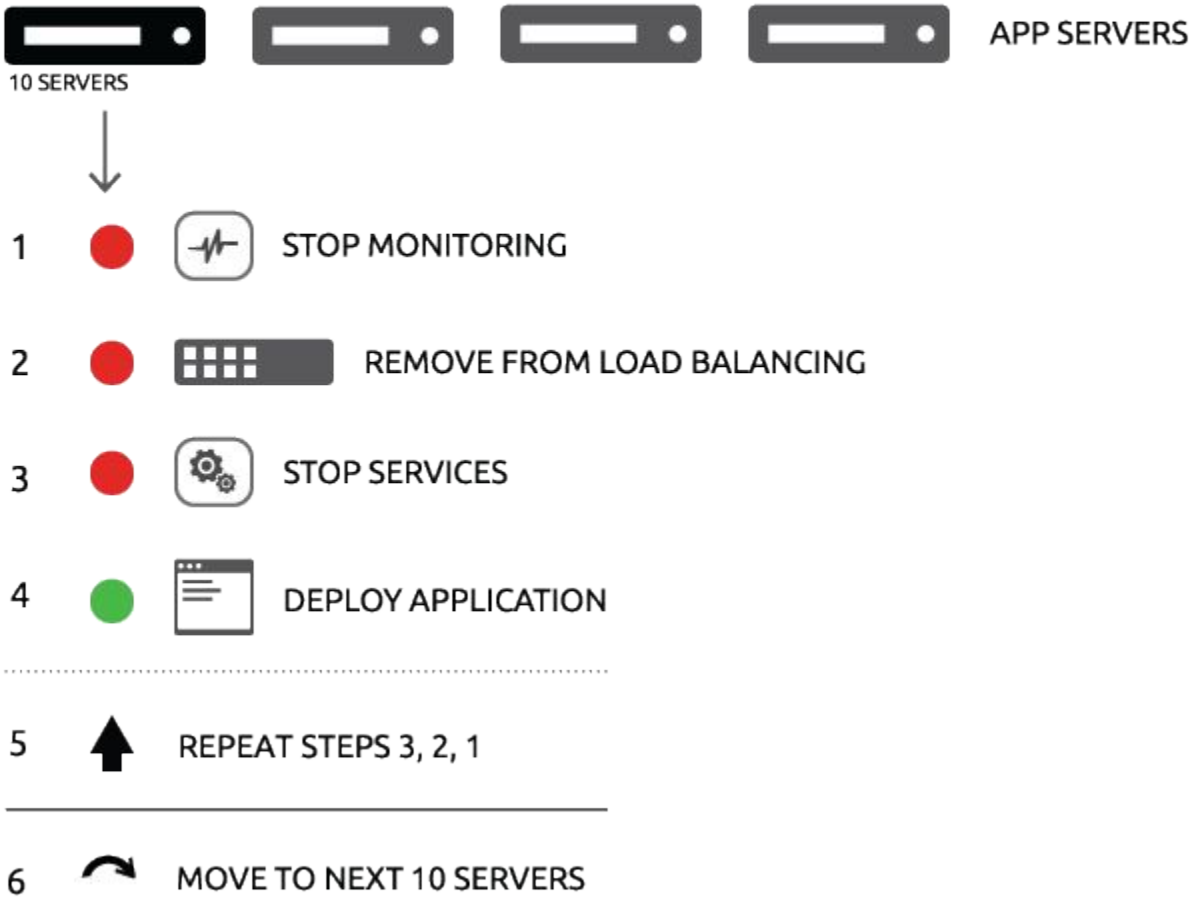
día 2

despliegue: ansible

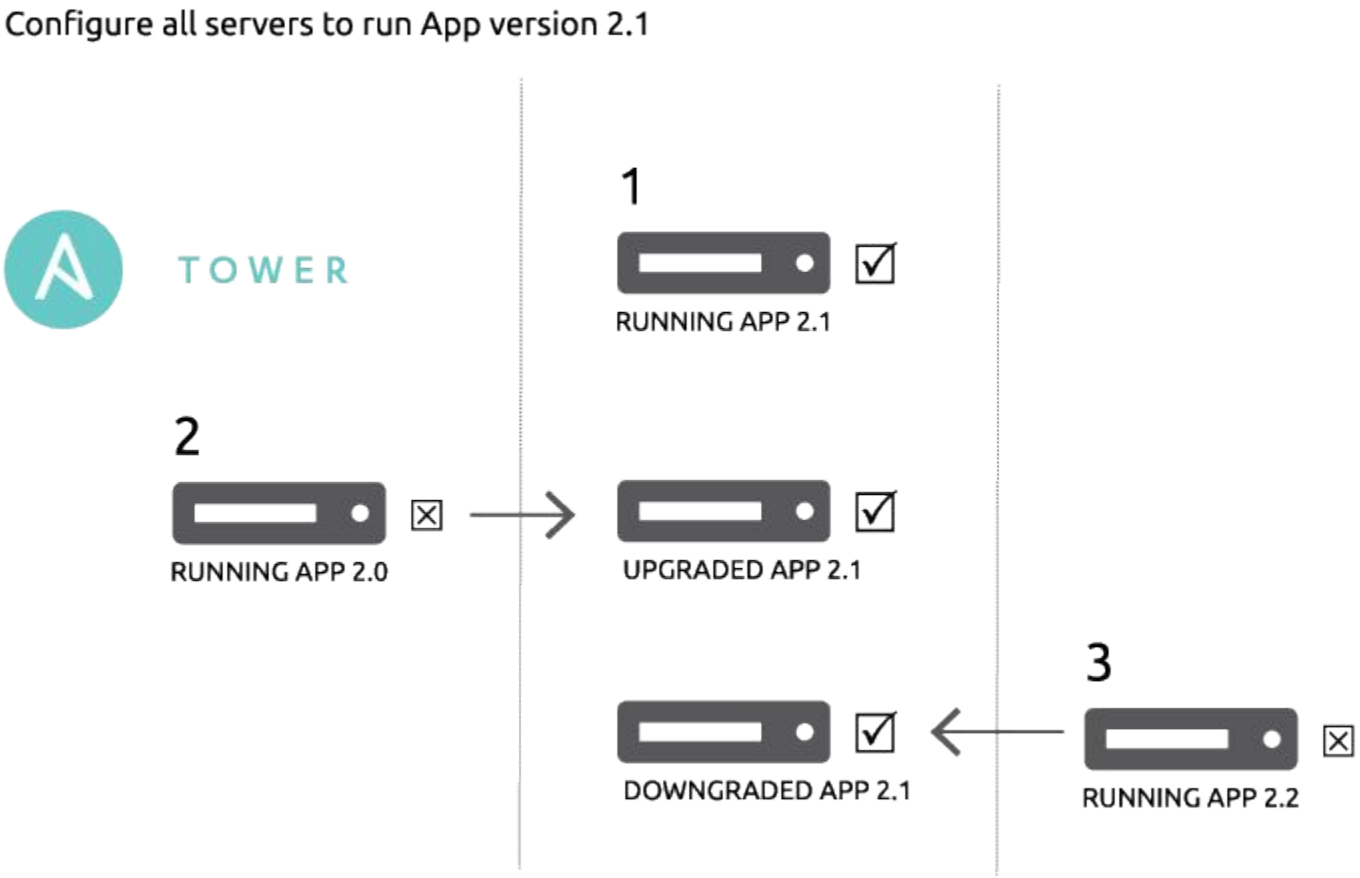
infraestructura como código
gestión de configuración

gestión de la configuración: herramientas

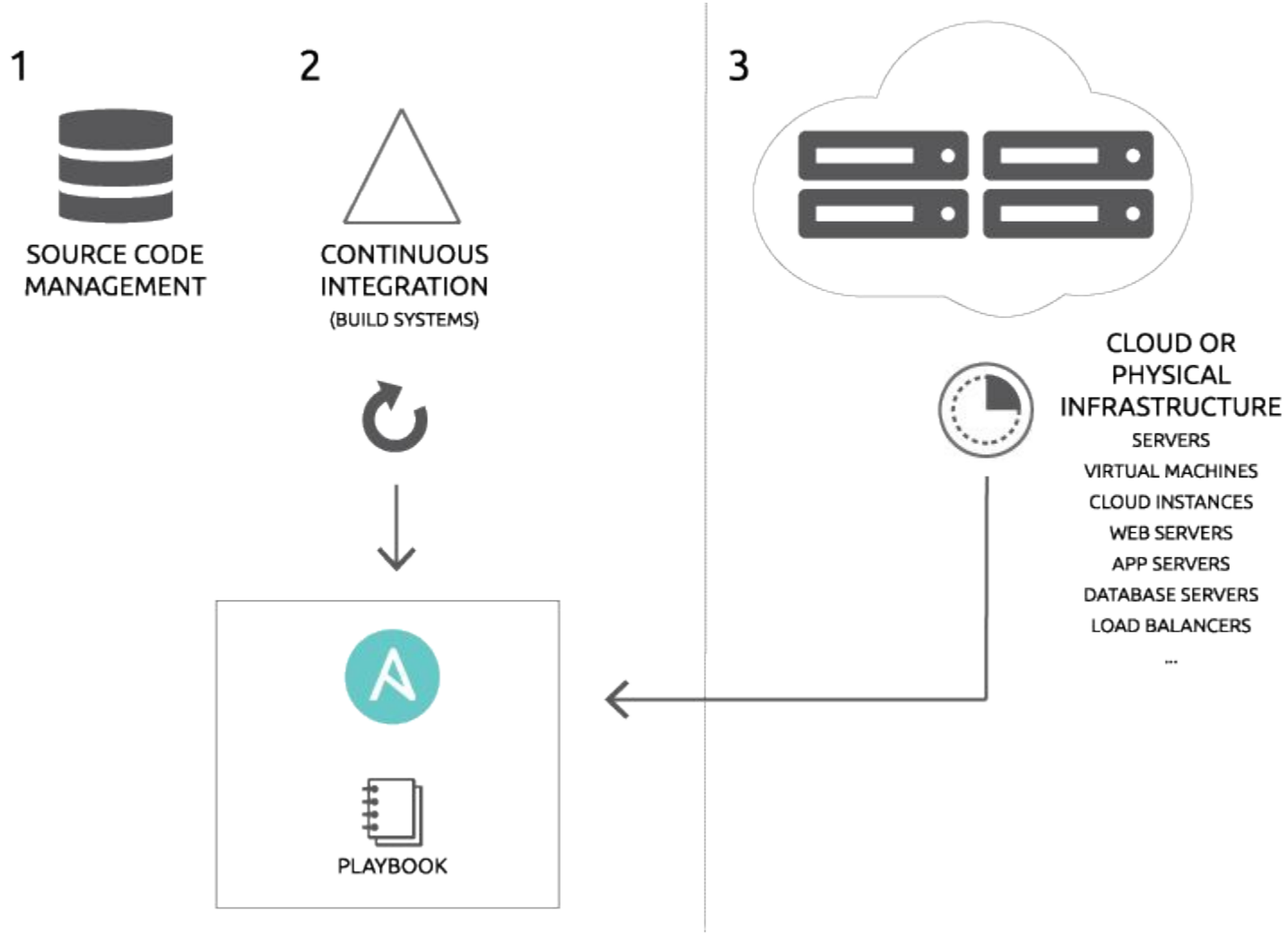
gestión de la configuración: ansible - application deployment



gestión de la configuración: ansible - configuration management



gestión de la configuración: ansible - continuous delivery



gestión de la configuración: ansible - práctica

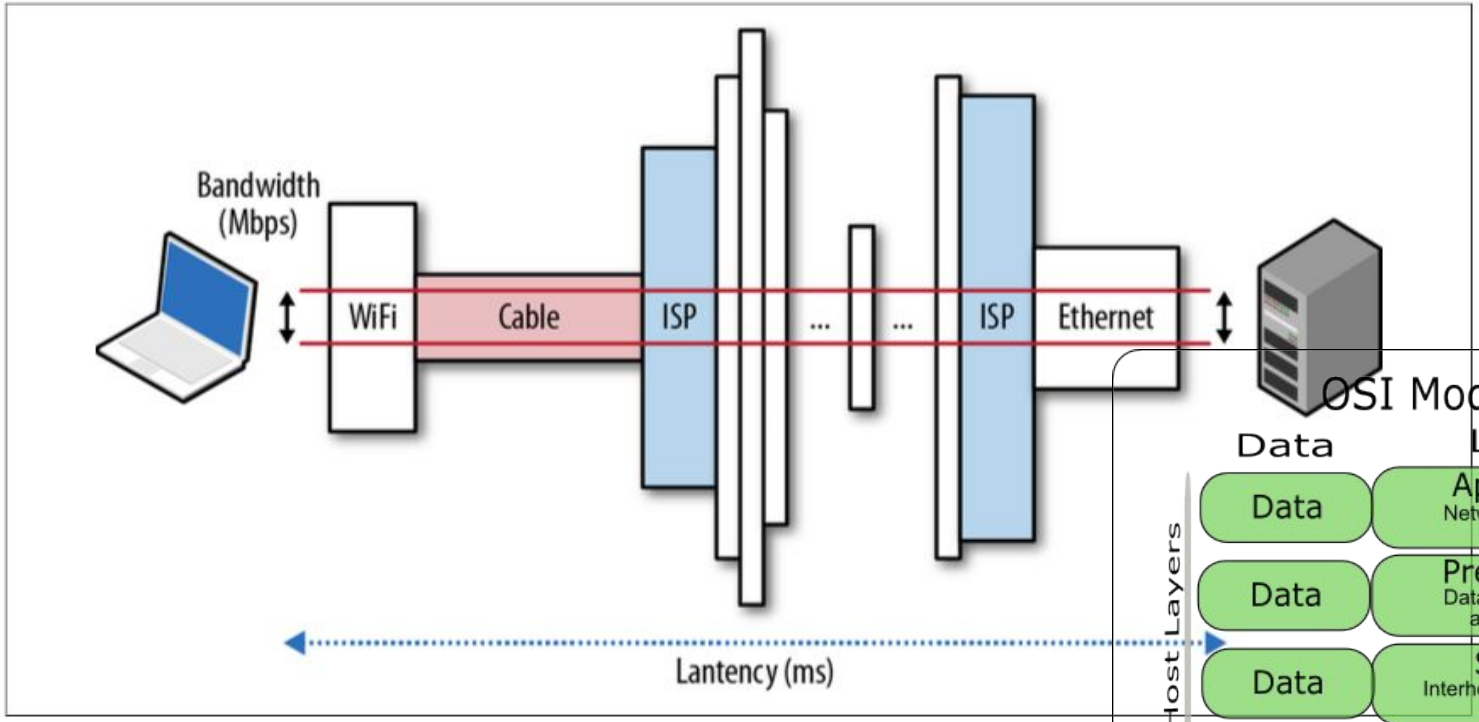
https://github.com/carlessanagustin/AITM

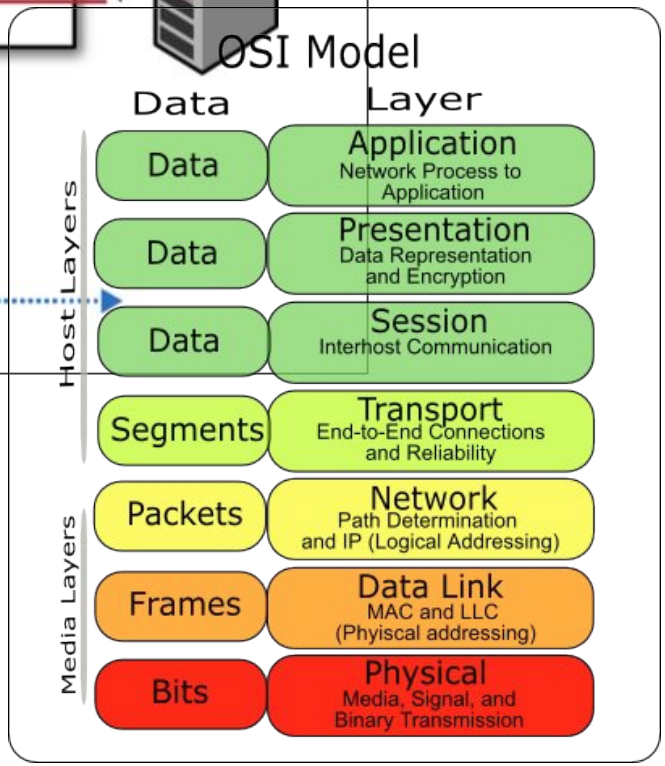
despliegue: rendimiento

rendimiento: herramientas

SilkPerformer
BlazeMeter WebLOAD
VisualStudioUltimate
NeoLoad loadUI Gatling
Blitz LoginVSI
JMeter Loaderio
LoadRunner OpenSTA
CloudTest
TestStudio

rendimiento web: arquitectura





rendimiento web: ejemplos

despliegue: nagios

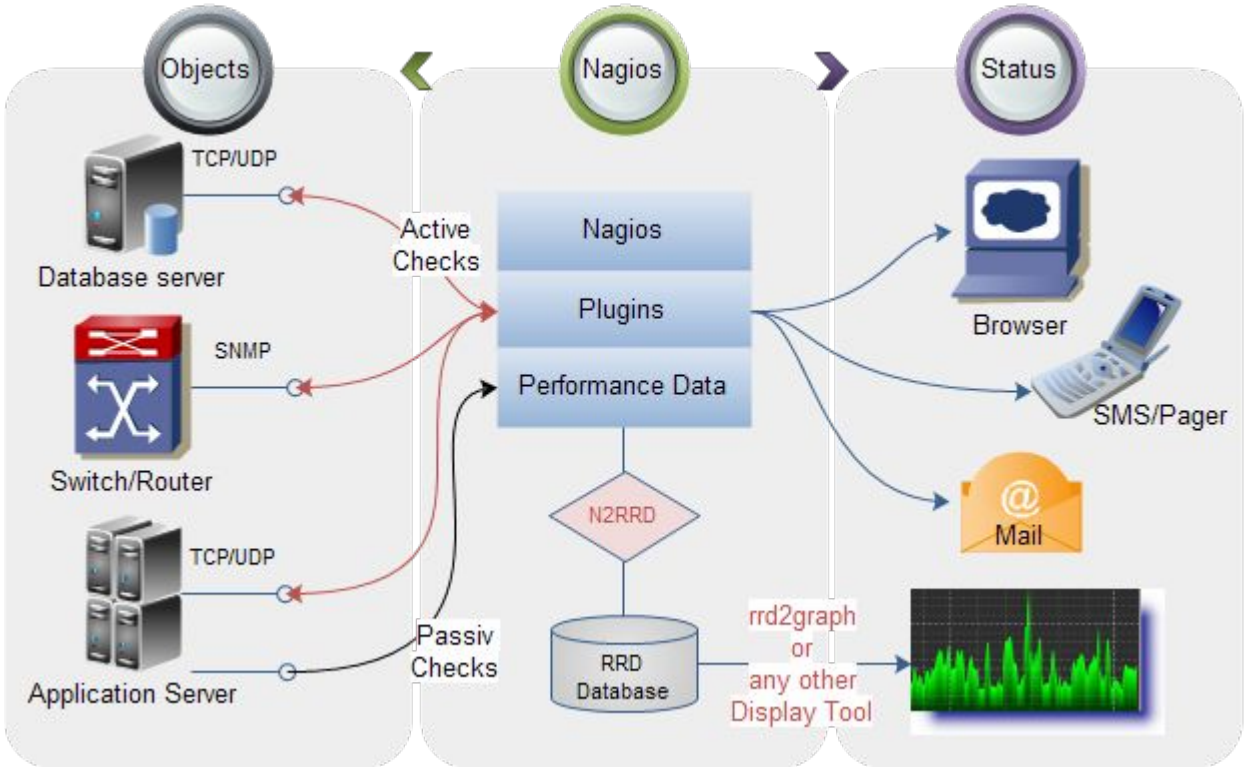
monitorización

monitorización: herramientas

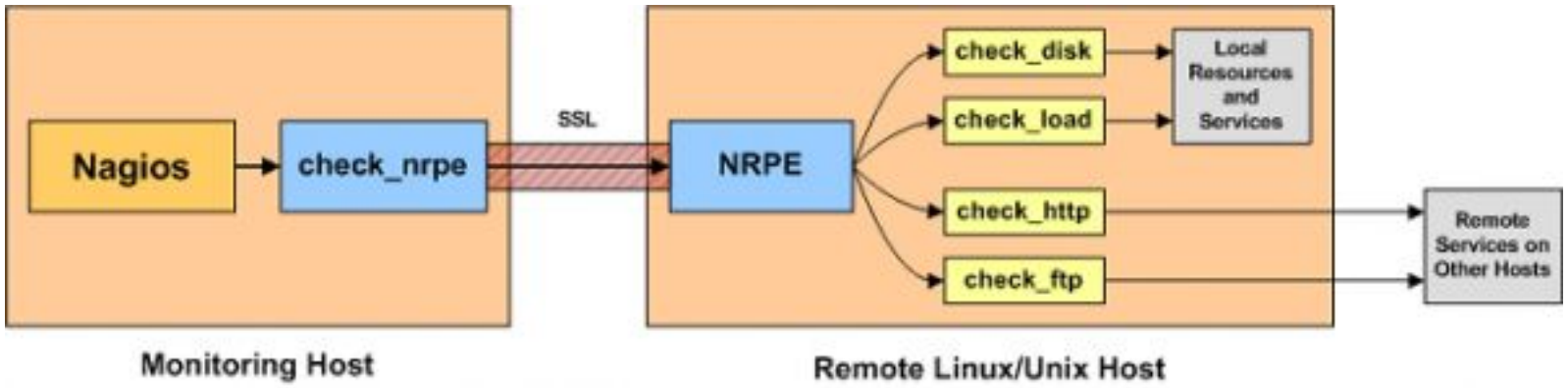
monitorización: tipos

- Monitoreo de sistemas
- Monitoreo de redes
- Monitoreo de la dependencias
- Integración
- Monitoreo de la actividad de negocio
- Instrumentación de procesos o traceo
- Procesamiento de eventos complejos

monitorización: nagios - arquitectura



monitorización: nagios remote plugin executor



monitorización de sistemas y redes: nagios - práctica

https://github.com/carlessanagustin/AITM

ejemplos reales

día 2

ejemplo 3.1: Andy Sykes on monitoring @ forward3d

- sensu: http://sensuapp.org/
- graphite: http://graphite.readthedocs.org/en/latest/
- flapjac: http://flapjack.io/

fuente: https://www.youtube.com/watch?v=Q9BagdHGopg

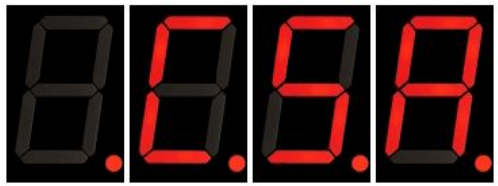
ejemplo 2.2: ruby on rails stack (sysdivision)

- ubuntu
- proxmox
- nginx
- unicorn
- ruby + rails
- capistrano
- redis
- mongodb
- capybara + cucumber

fuente: http://sysdivision.com/

Carles San Agustín

www.carlessanagustin.com



@carlesanagustin

linkedin.carlessanagustin.com

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
School of Professional & Executive Development

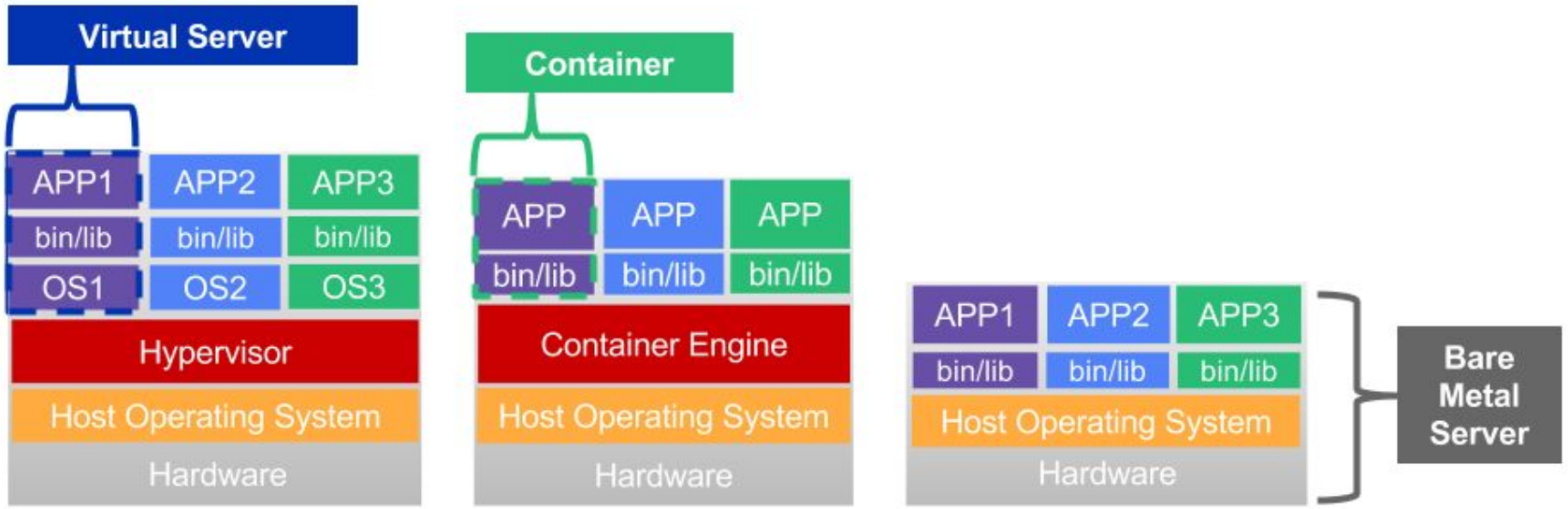
AGILE IT MANAGEMENT
SCRUM, **DEVOPS**, LEAN IT

prácticas de devops

entornos

día 3

entornos actuales



entornos: docker

contenedores de aplicaciones

separación de intereses

desarrolladores *o programadores*

- Código
- Librerías
- Aplicación
- Gestor de paquetes
- Información y datos

operaciones *o sysadmin*

- Registro (logging)
- Acceso remoto
- Configuración de red
- Monitorización

separación de intereses

desarrolladores *o programadores*

- Código
- Librerías
- Aplicación
- Gestor de paquetes
- Información y datos

operaciones *o sysadmin*

- Registro (logging)
- Acceso remoto
- Configuración de red
- Monitorización

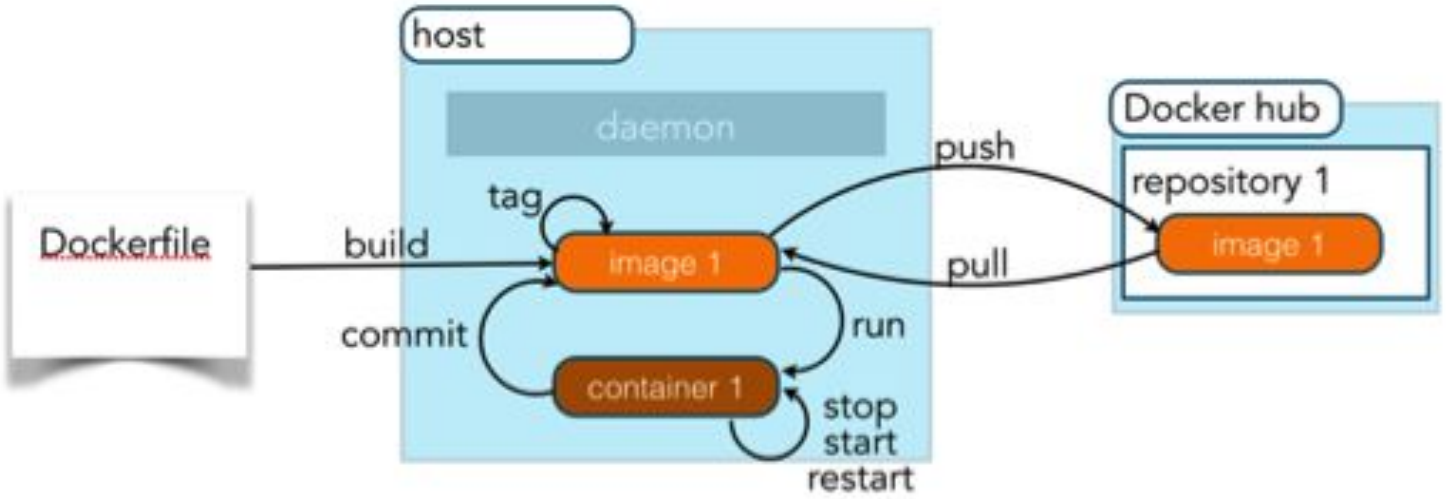
Consistencia de sistemas

Gestión de redes y volúmenes nativo

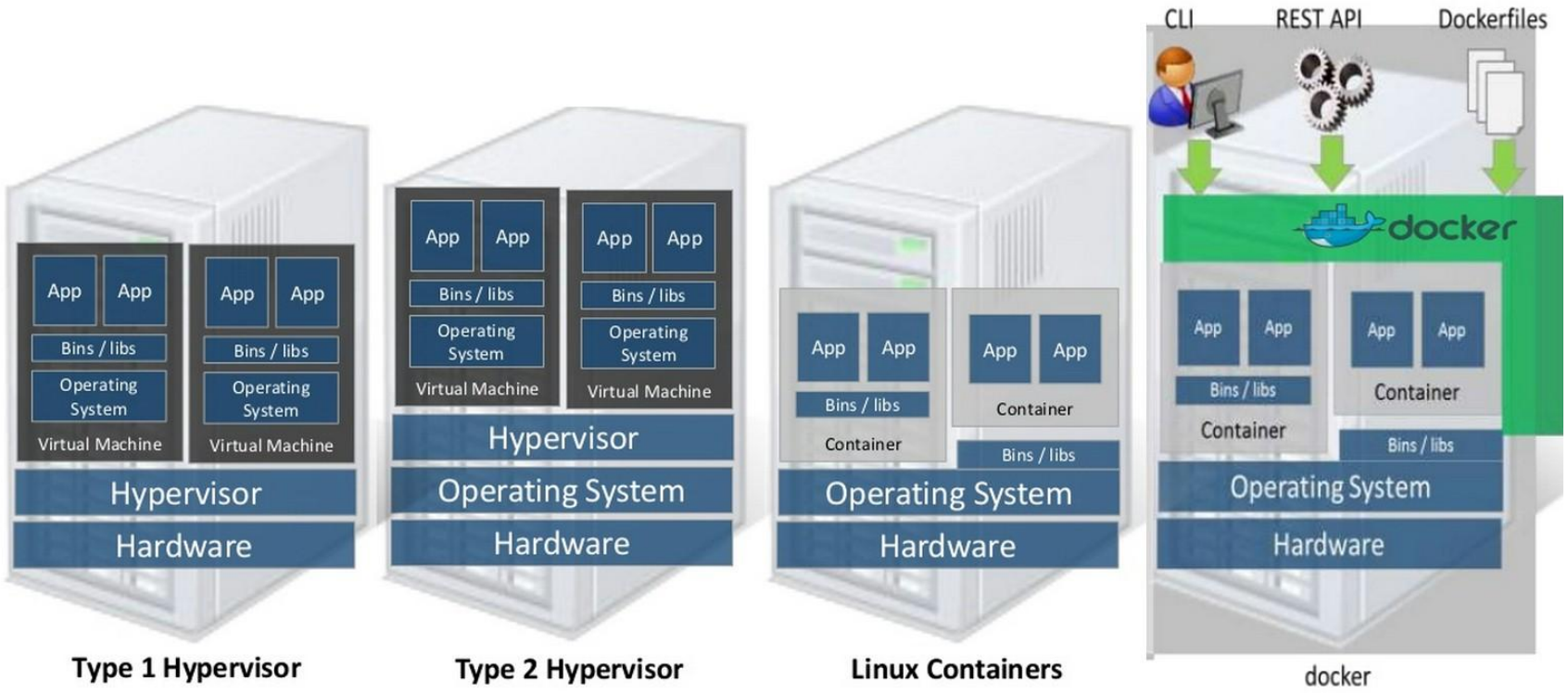
Clustering y redes escalable nativo

Flexibilidad con arquitectura conectable al ecosistema

docker: workflow



máquinas virtuales vs **docker**



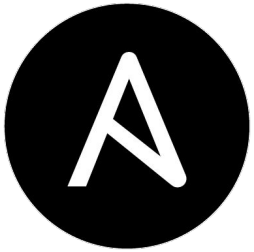
Oracle VM Server, Citrix XenServer, VMware ESX/ESXi, Microsoft Hyper-V, ...

VMware Workstation, VMware Player, VirtualBox, Parallels, KVM, QEMU, ...

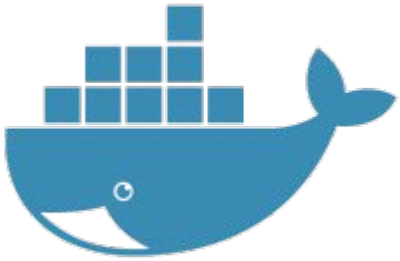
LXC

Docker

ansible vs **docker**

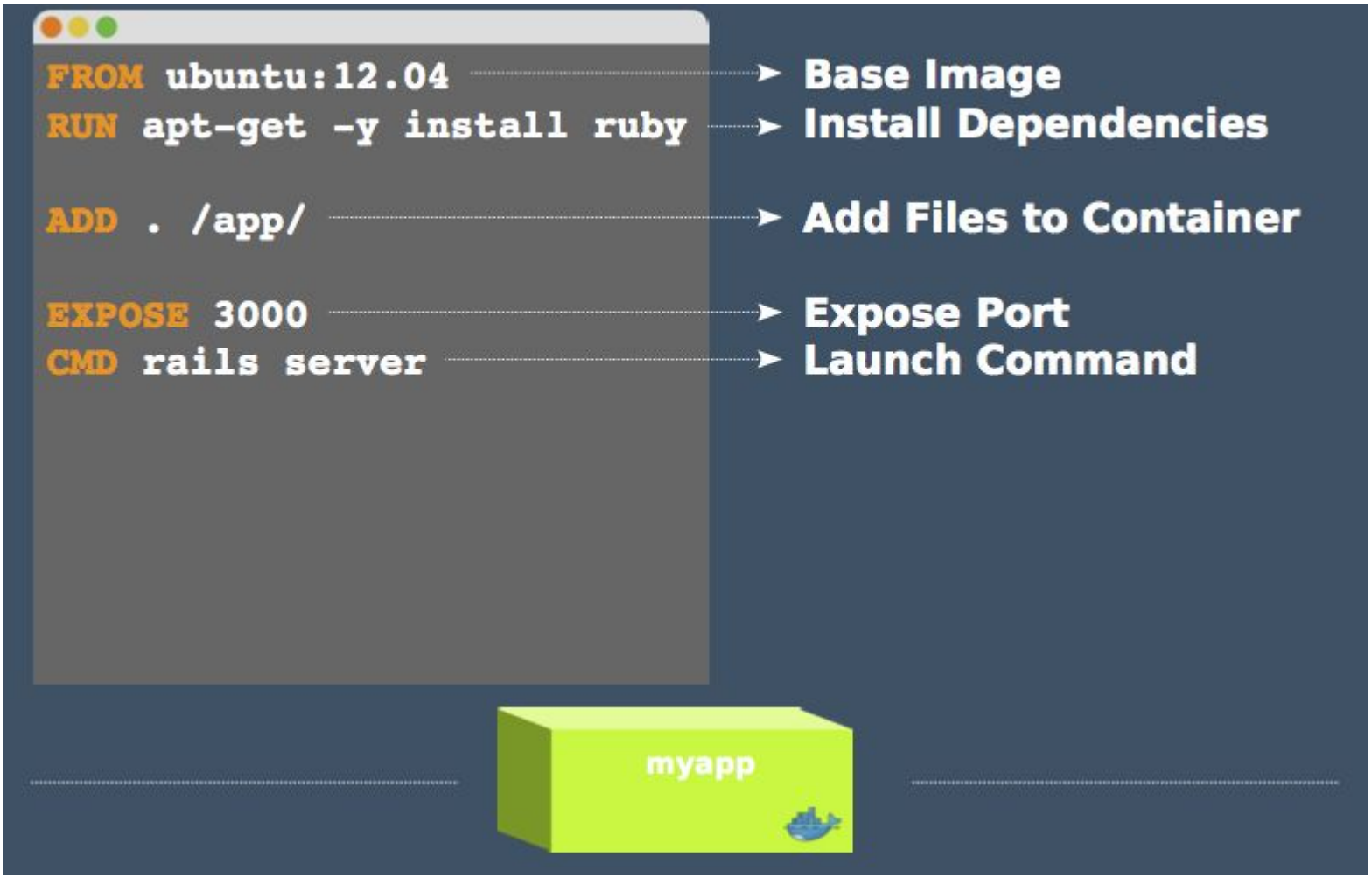


- Muchas veces depende de otros playbooks o repositorios (como apt-get o yum) para instalar software.
- Las dependencias son resueltas a tiempo de convergencia (instalación o actualización)
- Los cambios pueden ser desplegados para evolucionar un servidor.



- Docker aprovisiona aplicaciones en servidores en una capa separada del sistema operativo.
- Las dependencias son resueltas en el momento de crear la imagen (ejemplo: *docker build o docker entrypoint*).
- Los cambios deben hacerse deteniendo y sustituyendo un container por una nueva versión (matar el ganado).

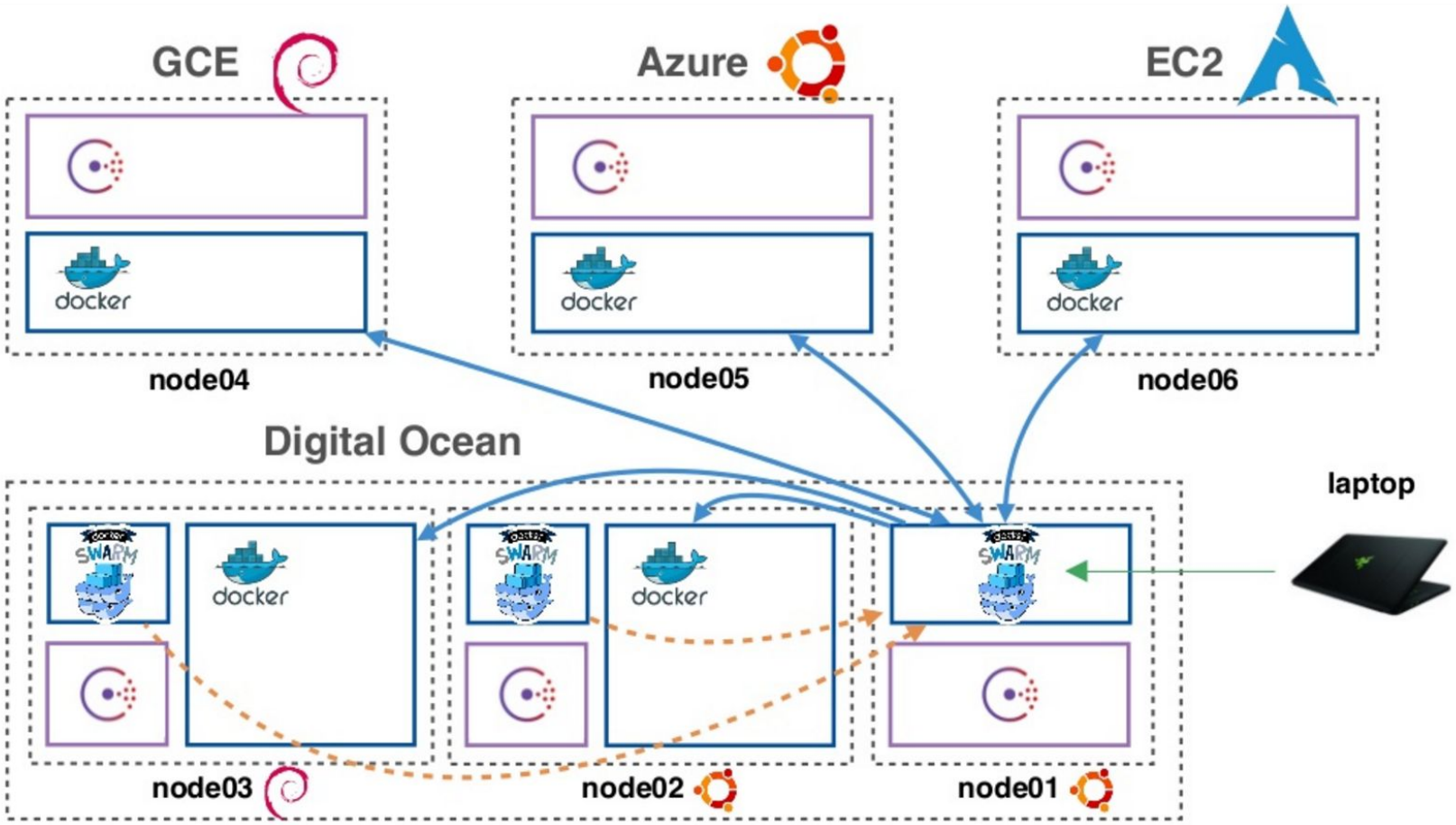
docker: dockerfile



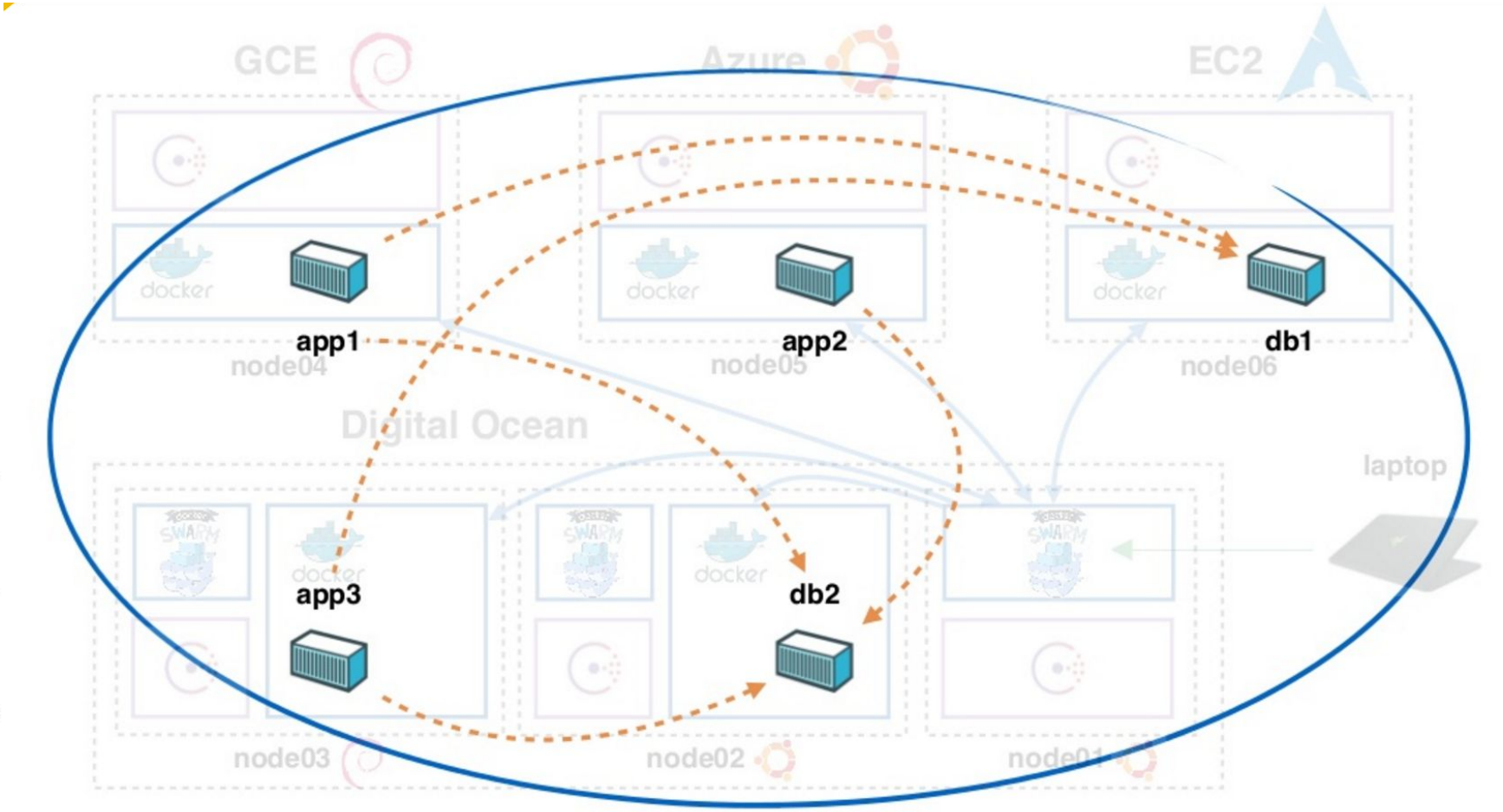
construye, empaqueta y ejecuta cualquier aplicación, en cualquier lugar

Docker motto: Build, Ship, and Run Any App, Anywhere

docker: sistemas distribuidos

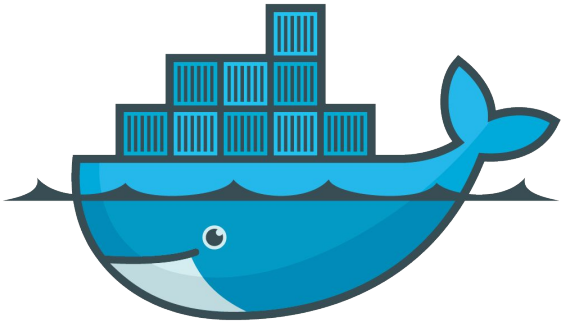


docker: sistemas distribuidos



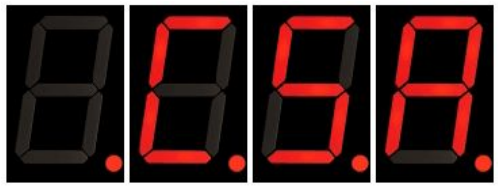
entornos:
docker - práctica

https://github.com/carlessanagustin/AITM



Carles San Agustín

www.carlessanagustin.com



@carlesanagustin

linkedin.carlessanagustin.com