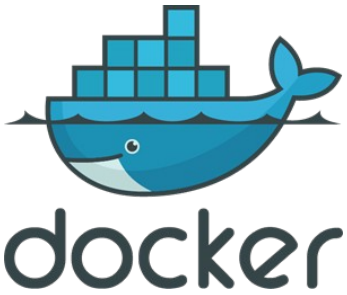
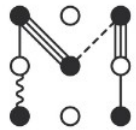
PHP e DevOps

Build a complete CI/CD pipeline



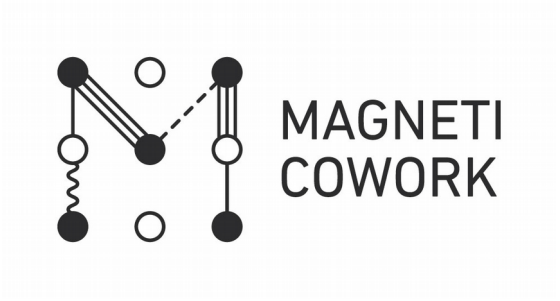
26/04/2018
Palermo



MagnetiCowork
Via E. Amari
148

Organized by

http://palermo.grusp.org



whoami

whoami

Ops/DevOps with high security trends & dev skills.
His passion was born approaching reality like the FreakNet CT and co-founding hacklab in Palermo.
With a background as IT Specialist Consultant at Unicredit Group, he currently deals with the management and automation of networks&systems and development for Viral Digital Strategies and other realities.
Member of the FablabPalermo and MUSIF, he firmly believes in open source and the "source code as a means of personal and social evolution".
His motto?

- ***"Talk is cheap. Show me the code."***
  (Linus Torvalds)

Intro

$ assertTrue

- **LINUX [shell, ssh, basic scripting ecc...]**
- **GIT [must]**
- **Docker [basic]**
- **PHP [ahah]**
- **SQL [basic]**
- **PHP-FIG [psr and more]**
- **Composer&Packagist**
- **Will&Curiosity&Determination**
- **English! [the IT language: resistance is futile]**

$ assertTrue

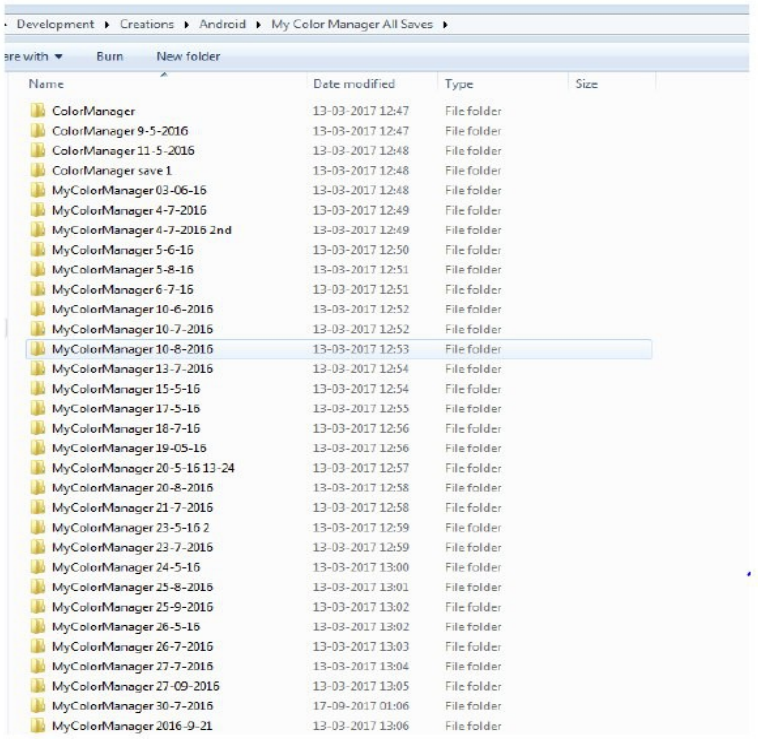
WHY GIT???

Who would win

Most advanced version control system used to keep track of changes in any set of files. Aimed at speed, data integrity, and support for distributed, non-linear workflows.

git

Me



$ assertTrue

GIT IS HARD???

$ man man

- **Slideshow**
- **Git repository**
  - **https://git{lab|hub}.com/gionniboy/php-democicd**
- **Fb/Telegram groups**
- **Reference**
- **Don’t be shy – ask to community [rtfm first]**

Continuous Integration

man ci

Continuous Integration is a software development practice where members of a team integrate their work frequently, usually each person integrates at least daily - leading to multiple integrations per day. Each integration is verified by an automated build (including test) to detect integration errors as quickly as possible. *(Martin Fowler)*

man ci

- **Software development practice**
- **Software is always integrated**
- **Software is always in a working state ready to run.**
- **If something breaks software: we know what and where**
- **Quick fix of broken parts**

Pipeline Failure → Fix

Pipeline Success → Software is ready to be delivered

CI Practices

- **AVOID LEAK! BE CAREFUL!**
- **Maintain a code repository**
- **Automate the build**
- **Make the build self-testing**
- **Everyone commits to the baseline every day**
- **Every commit (to baseline) should be built**
- **Keep the build fast**
- **Everyone can see the results of the builds**
- **Master is GOD**

$ man ci

CI Benefits

- **Risk Mitigation** (no more: it worked on my machine)
- **Confidence** (move fast, break things, fix it. Repeat.)
- **Team Communication & Documentation** (keep all updated)
- **Reduced Overhead** (code review, manual task repetition)
- **Consistency of Build Process** (same way for every change)

Continuous Delivery

$ git push --force origin master

$ man cd

Continuous Delivery is a software development discipline where you build software in such a way that the software can be released to production at any time.

You're doing continuous delivery when:

- Your software is deployable throughout its lifecycle
- Your team prioritizes keeping the software deployable over working on new features
- Anybody can get fast, automated feedback on the production readiness of their systems any time somebody makes a change to them
- You can perform push-button deployments of any version of the software to any environment on demand

$ man cd

You achieve **Continuous Delivery** by continuously integrating the software done by the development team, building executables, and running automated tests on those executables to detect problems. Furthermore you push the executables into increasingly production-like environments to ensure the software will work in production. To do this you use a DeploymentPipeline.

To achieve continuous delivery you need:

- a close, collaborative working relationship between everyone involved in delivery.
- extensive automation of all possible parts of the delivery process, usually using a DeploymentPipeline

(Martin Flower)

$ [[“cdelivery” != “cdeploy”]]

Continuous Delivery is sometimes confused with **Continuous Deployment**.

Continuous Deployment means that every change goes through the pipeline and automatically gets put into **Production**, resulting in many production deployments every day.

Continuous Delivery just means that you are able to do frequent deployments but may choose not to do it, usually due to businesses preferring a slower rate of deployment.

In order to do **Continuous Deployment** you must be doing **Continuous Delivery**.

(Martin Fowler)

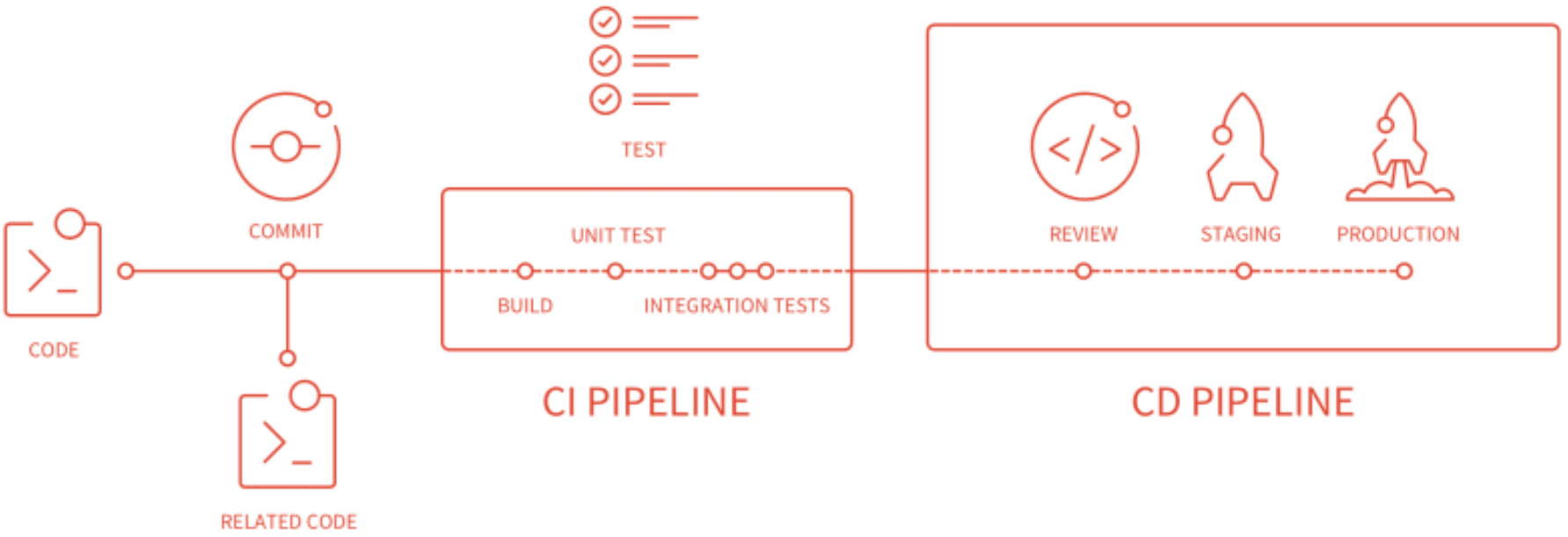
CD Benefits

Continuous Integration usually refers to integrating, building, and testing code within the development environment. **Continuous Delivery** builds on this, dealing with the final stages required for **Production Deployment**.

- **Reduced Deployment Risk** (smaller change, under hard control)
- **Believable Progress** (track all changes, turn back easily)
- **User Feedback** (quick feedback, release for subset users)

Amazing Pipeline

$ git rebase --onto master ci cd



PHP Analisys

$ echo “php anal”

- **PHPUNIT testing**
- **PHPCS code sniffer - linting**
- **PHPMD mess detector**
- **PHPMETRICS static analisys**
- **PHPSTAN static analisys**

Fear, Uncertainity, Doubt

References

https://gitlab.com/gionniboy/phpdemo-cicd
http://palermo.grusp.org
https://t.me/phpugpa
https://fb.com/groups/PalermoPUG
https://www.docker.com/
https://gitlab.com
http://yaml.org/
https://www.php-fig.org/psr/
https://git-scm.com/
https://getcomposer.org/
https://github.com/sebastianbergmann/phpunit
https://github.com/squizlabs/PHP_CodeSniffer
https://github.com/phpmd/phpmd
https://github.com/phpmetrics/PhpMetrics
https://github.com/phpstan/phpstan
https://github.com/guzzle/guzzle
https://www.martinfowler.com/articles/continuousIntegration.html
https://martinfowler.com/bliki/ContinuousDelivery.html
https://martinfowler.com/bliki/BlueGreenDeployment.html
http://www.javatechblog.com/programming/funny-source-code-comments-by-programmers/

echo ~gionni

print(“You can contact {} on”.format(“G.B. Pullarà”))

giovbat@gmail.com
info@firegarden.co

https://www.linkedin.com/in/firegarden/

Or open issues/PR on demo repos

https://gitlab.com/gionniboy/phpdemo-cicd

So Long, and Thanks for All the Fish

PHP User Group Palermo