italiaonline

Continuous Delivery

Continuous Integration

Marco Ghisellini – Assago, 19/04/2019

I nostri brand

LIBERO VIRGILIO PgCasa.it SUPEREVA DiLei SìViaggia QuiFinanza Buonissimo PagineGialle PagineBianche TuttoCittà

Summary

- What “Continuous Integration” and “Continuous Delivery” mean?
- What’s wrong with the common way of delivery?
- Which tools are used for automating the process?
- Sample pipeline
- Demo
- Releasing strategy

italiaonline

What “Continuous Integration” and “Continuous Delivery” mean?

LIBERO

Il numero uno dei portali italiani

I nostri brand

LIBERO VIRGILIO PgCasa.it SUPEREVA DiLei SìViaggia QuiFinanza Buonissimo Pagine Gialle Pagine Bianche Tutto Città

Continuous Integration (CI)

…Continuous Integration (CI) is the process of automatically building software

after new bits of code are integrated into a shared repository.

This yields “builds” of the code base that are in a working state at all times….

From Pivotal

Continuous Delivery (CD)

…Continuous Delivery (CD) is the process of automatically deploying the artifacts created with CI. [Tools]...are used to provide a consistent method of deploying software and configuration changes into an environment. The consistency CD provides means production deployments become a non-event….

From Pivotal

italiaonline

What’s wrong with the common way of delivery ?

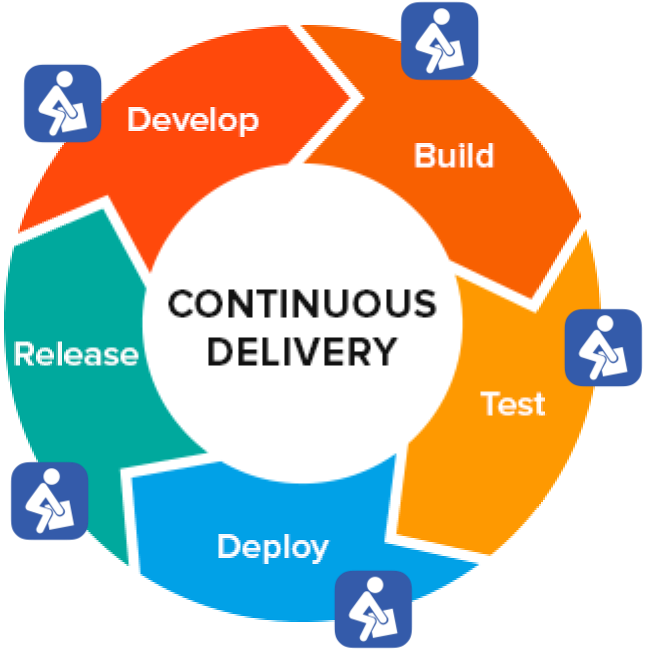
V:RGILIO

Il primo portale internet nato in Italia

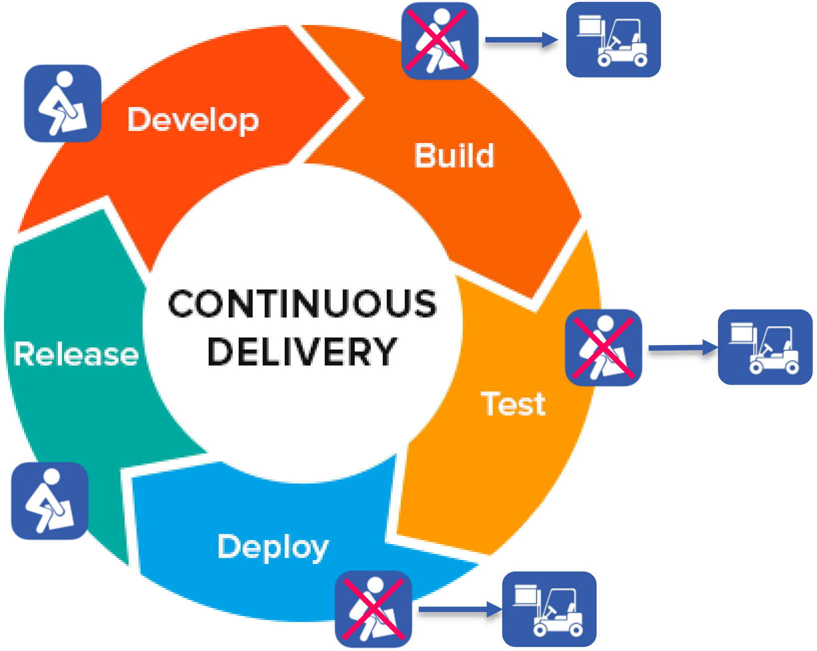
I nostri brand

LIBERO V:RGILIO PgCasa.it SUPEREVA Di·Lei SìViaggia QuiFinanza Buonissimo Pagine Gialle Pagine Bianche Tutto Città

Code lifecycle - Handmade



Code lifecycle - Automatic



Issues with Team collaboration

- Produce a release without including all the files pushed;
- Repository’s push which breaks the solution integrity;
- Produce a release which does not satisfy all the suit tests (unit and integration);
- Difficult to reproduce the test environment on local machine;
- Keep the test environment up & running and time consistency;
- …Generally, execute all these steps manually!!

Antipattern: Deploying Software Manually

- The production of extensive, detailed documentation that describes the steps to be taken and the ways in which the steps may go wrong;
- Reliance on manual testing to confirm that the application is running correctly;
- Frequent corrections to the release process during the course of a release;
- When the deployment process is not automated, it is not repeatable or reliable, leading to time wasted on debugging deployment errors;
- Manual deployments depend on the deployment expert;
- Performing manual deployments is boring and repetitive and yet needs significant degree of expertise;
- The only way to test a manual deployment process is to do it.

From “Continuous Delivery - Reliable Software Releases Through Build, Test And Deployment Automation”

italiaonline

Which tools are used for automating the process?

PgCasa.it

L’home service
su misura

Lavori da fare in casa?

I nostri brand

LIBERO VIRGILIO PgCasa.it SUPEREVA DiLei SìViaggia QuiFinanza Buonissimo PagineGialle PagineBianche TuttoCittà

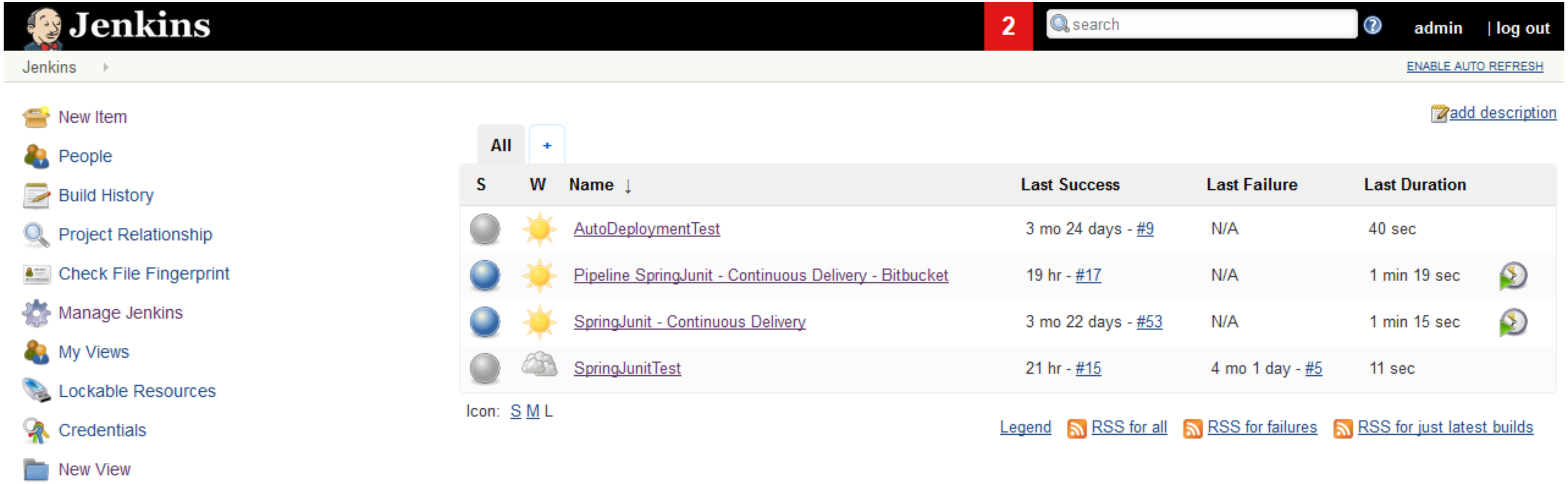
Tools

- Jenkins
- BitBucket
- Git
- Docker
- Maven

Jenkins

Jenkins is an open source automation server …

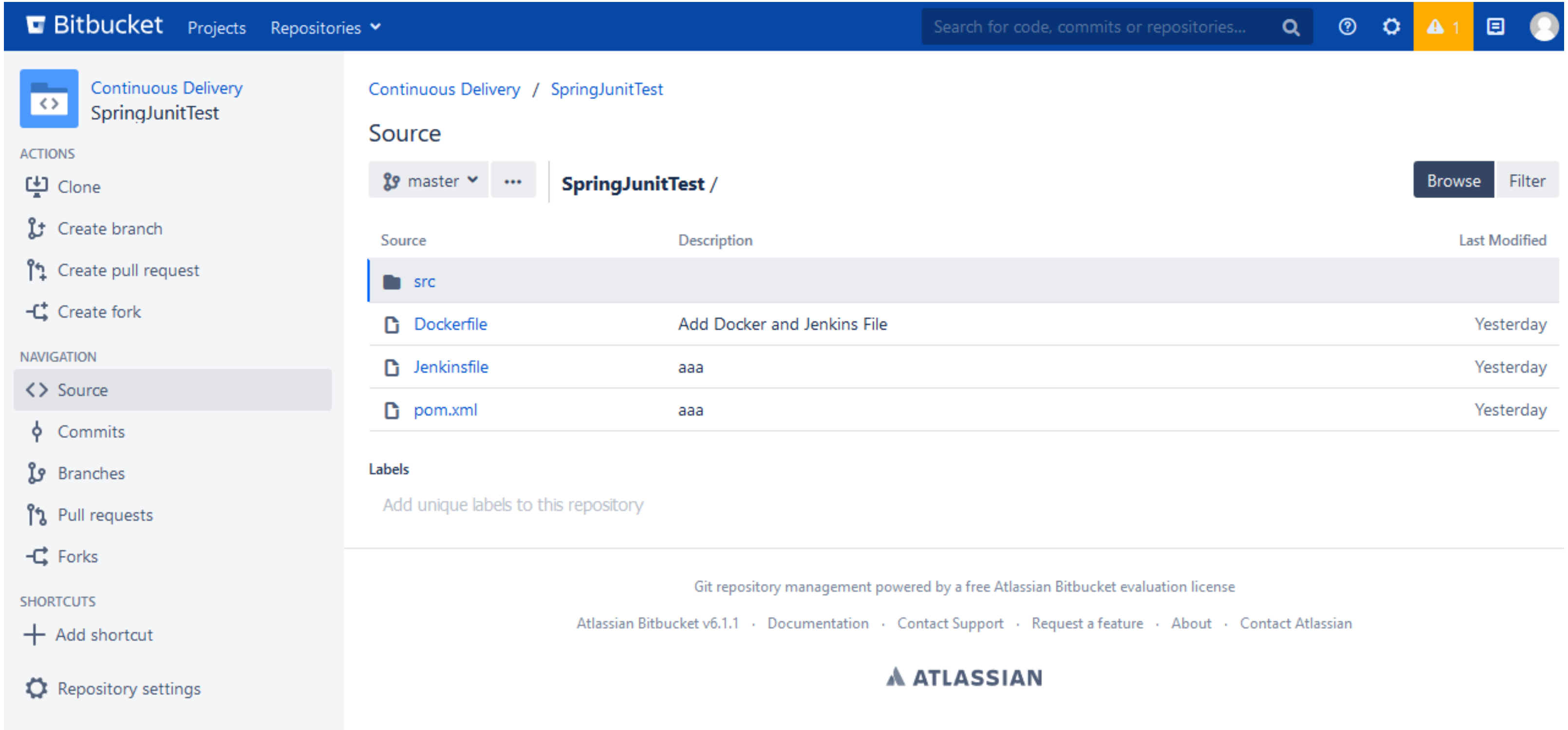
from Wikipedia



S	W	Name ↓	Last Success	Last Failure	Last Duration
●	☀	AutoDeploymentTest	3 mo 24 days - #9	N/A	40 sec
●	☀	Pipeline SpringJunit - Continuous Delivery - Bitbucket	19 hr - #17	N/A	1 min 19 sec
●	☀	SpringJunit - Continuous Delivery	3 mo 22 days - #53	N/A	1 min 15 sec
●	☁	SpringJunitTest	21 hr - #15	4 mo 1 day - #5	11 sec

Bitbucket

A git-based repository server.



Continuous Delivery / SpringJunitTest

Source

SpringJunitTest /

Source	Description	Last Modified
src		
Dockerfile	Add Docker and Jenkins File	Yesterday
Jenkinsfile	aaa	Yesterday
pom.xml	aaa	Yesterday

Labels

Add unique labels to this repository

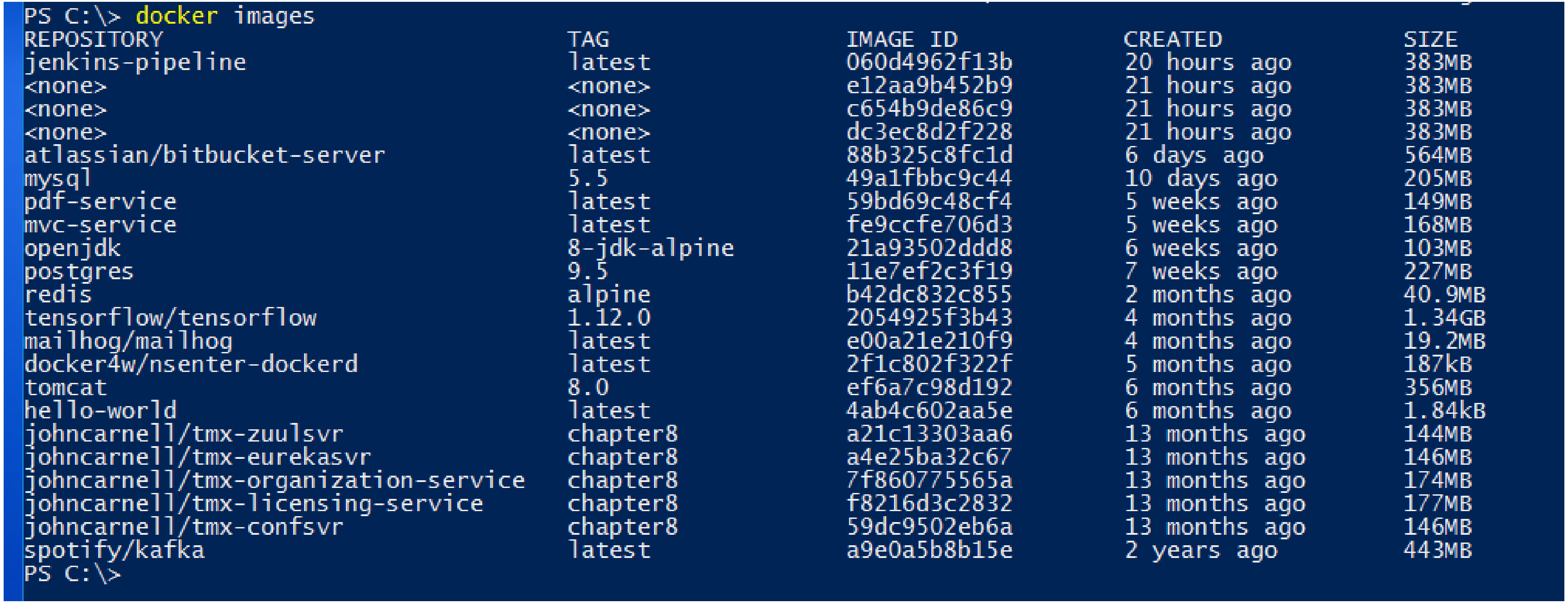
Git repository management powered by a free Atlassian Bitbucket evaluation license

Atlassian Bitbucket v6.1.1 · Documentation · Contact Support · Request a feature · About · Contact Atlassian

ATLASSIAN

Docker

An virtualiser application container.



```
PS C:\> docker images
REPOSITORY                              TAG                 IMAGE ID            CREATED             SIZE
jenkins-pipeline                        latest              060d4962f13b        20 hours ago        383MB
<none>                                  <none>              e12aa9b452b9        21 hours ago        383MB
<none>                                  <none>              c654b9de86c9        21 hours ago        383MB
<none>                                  <none>              dc3ec8d2f228        21 hours ago        383MB
atlassian/bitbucket-server              latest              88b325c8fc1d        6 days ago          564MB
mysql                                   5.5                 49a1fbbc9c44        10 days ago         205MB
pdf-service                             latest              59bd69c48cf4        5 weeks ago         149MB
mvc-service                             latest              fe9ccfe706d3        5 weeks ago         168MB
openjdk                                 8-jdk-alpine        21a93502ddd8        6 weeks ago         103MB
postgres                                9.5                 11e7ef2c3f19        7 weeks ago         227MB
redis                                   alpine              b42dc832c855        2 months ago        40.9MB
tensorflow/tensorflow                   1.12.0              2054925f3b43        4 months ago        1.34GB
mailhog/mailhog                         latest              e00a21e210f9        4 months ago        19.2MB
docker4w/nsenter-dockerd                latest              2f1c802f322f        5 months ago        187kB
tomcat                                  8.0                 ef6a7c98d192        6 months ago        356MB
hello-world                             latest              4ab4c602aa5e        6 months ago        1.84kB
johncarnell/tmx-zuulsvr                 chapter8            a21c13303aa6        13 months ago       144MB
johncarnell/tmx-eurekasvr               chapter8            a4e25ba32c67        13 months ago       146MB
johncarnell/tmx-organization-service    chapter8            7f860775565a        13 months ago       174MB
johncarnell/tmx-licensing-service       chapter8            f8216d3c2832        13 months ago       177MB
johncarnell/tmx-confsvr                 chapter8            59dc9502eb6a        13 months ago       146MB
spotify/kafka                           latest              a9e0a5b8b15e        2 years ago         443MB
PS C:\>
```

italiaonline

Sample pipeline

SUPEREVA

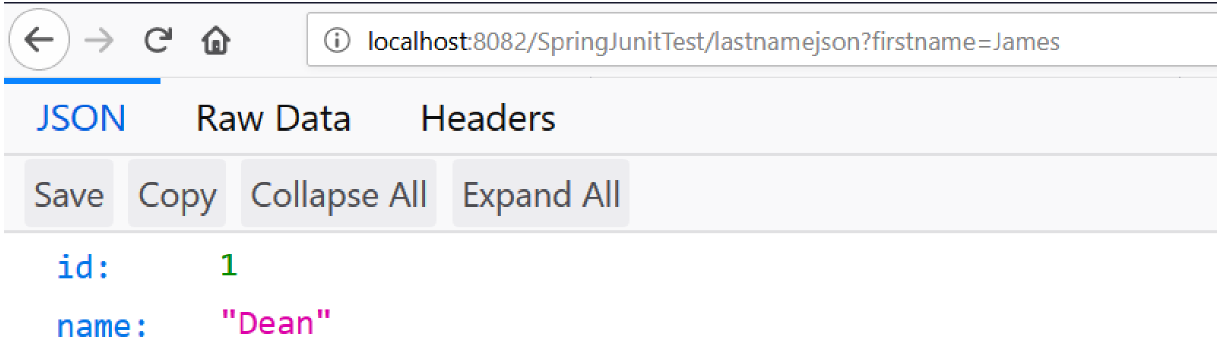
Quando le notizie
sono di tendenza

I nostri brand

LIBERO VIRGILIO PgCasa.it SUPEREVA DiLei SiViaggia QuiFinanza Buonissimo Pagine Gialle Pagine Bianche Tutto Città

Sample project – User registry

A web application which contains a user registry data (name, surname) and exposes the endpoint /lastnamejson?firstname=<James|Larry>.

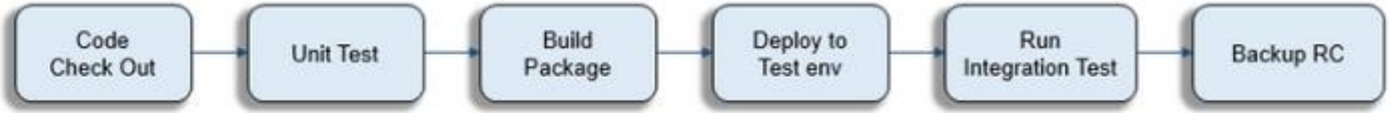


```
id:      1
name:    "Dean"
```

The web application is deployed into Tomcat 8 and the data is stored into a MySql Database.

Continuous delivery - User registry/2

- Every push to repository triggers the execution of the above pipeline on Jenkins;
- The first error raised stopped the pipeline execution.



Jenkins – Pipeline/1

```
def CONTAINER_NAME="jenkins-pipeline"
def CONTAINER_TAG="latest"
def HTTP_PORT="8082"
def TEST_URL="http://localhost:${HTTP_PORT}/SpringJunitTest/lastnamejson?firstname=James"
def TEST_TEXT="{\"id\":1,\"name\":\"Dean\"}"

node {

    stage('Checkout') {
            checkout scm
    }
    stage('Unit Test'){
        bat "mvn -U clean compile"
    }

     stage('Package'){
        bat "mvn package -DskipTests"
    }
```

Jenkins – Pipeline/2

```
stage("Image Clear up"){
        try {
        bat "docker stop mysql"
        bat "docker rm -f mysql"
        bat "docker stop $CONTAINER_NAME"
        bat "docker rm -f $CONTAINER_NAME"
        bat "docker rmi $CONTAINER_NAME:$CONTAINER_TAG"
    } catch(error){}
  }

  stage('Image Build'){
    bat "docker build -t $CONTAINER_NAME:$CONTAINER_TAG  -t $CONTAINER_NAME --pull --no-cache ."
    echo "Image build complete"
  }

  stage('Run App'){
    bat "docker run --name mysql -p 3306:3306 -e MYSQL_ALLOW_EMPTY_PASSWORD=yes -d mysql:5.5"
    sleep(10)
    bat "docker run -d -p $HTTP_PORT:8080 --link mysql:mysql --name $CONTAINER_NAME $CONTAINER_NAME:$CONTAINER_TAG"
    echo "Application started on port: ${HTTP_PORT} (http)"
  }
```

Jenkins – Pipeline/3

```
 stage('Integration Test'){
        sleep(10)
        bat "docker exec -i mysql mysql -e \"create schema junit;use junit;DROP TABLE IF EXISTS Customers;CREATE TABLE Customers (  id INT NOT NULL
AUTO_INCREMENT,  first_name VARCHAR(30),  last_name  VARCHAR(30),  PRIMARY KEY (`id`));INSERT INTO Customers VALUES (1, 'James', 'Dean');INSERT
INTO Customers VALUES (2, 'Larry', 'Bird');\"«

        def response = httpRequest "${TEST_URL}"
        println("Status: "+response.status)
        println("Content: "+response.content)
        if (!response.content.contains("${TEST_TEXT}"))
        {
            echo '[FAILURE] Failed to verify TEST_TEXT'
            currentBuild.result = 'FAILURE'
            return
        }
    }
    stage('Backup Release'){
        fileOperations([fileCopyOperation(excludes: '', flattenFiles: true, includes: 'target/SpringJunitTest.war', targetLocation: 'C:\\deploy')])
        fileOperations([fileRenameOperation(destination: 'C:\\deploy\\SpringJunitTest-$BUILD_TIMESTAMP.war', source: 'C:\\deploy\\SpringJunitTest.war')])
    }
}
```

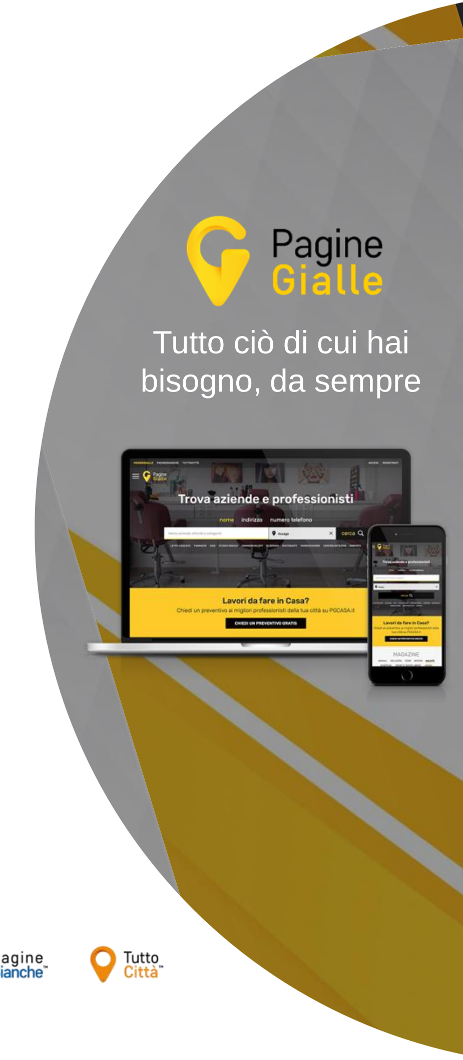
Docker – Containers

```
REPOSITORY          TAG          IMAGE ID          CREATED          SIZE
jenkins-pipeline      latest         060d4962f13b      24 hours ago       383MB
mysql                 5.5          49a1fbbc9c44      10 days ago        205MB

CONTAINER ID    IMAGE                   COMMAND                      CREATED          STATUS            PORTS                        NAMES
2ebaee5881eb    jenkins-pipeline:latest  "catalina.sh run"            24 hours ago     Up 3 seconds      0.0.0.0:8082->8080/tcp       jenkins-pipeline
102680c1bc24   mysql:5.5                "docker-entrypoint.s…"       24 hours ago     Up 33 seconds     0.0.0.0:3306->3306/tcp       mysql
```

Demo!

https://github.com/MarcoGhise/SpringJunitTest/blob/master/ContinuousDelivery.mp4


Pagine Gialle
Tutto ciò di cui hai bisogno, da sempre
Trova aziende e professionisti
Lavori da fare in Casa?
CHIEDI UN PREVENTIVO GRATIS

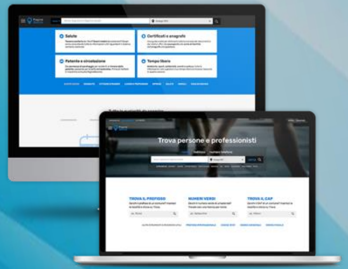
italiaonline

Releasing strategy

Blue-green releasing
Canary releasing (dark launches)
Shadowing Traffic releasing

Pagine Bianche

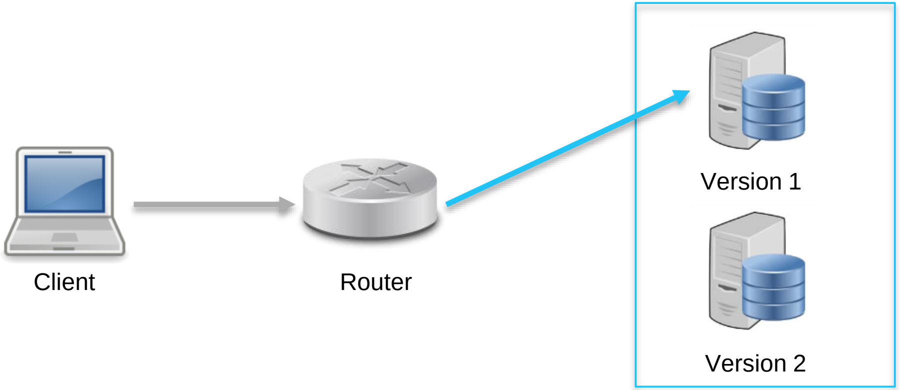
Un tesoro di
10 milioni di abbonati



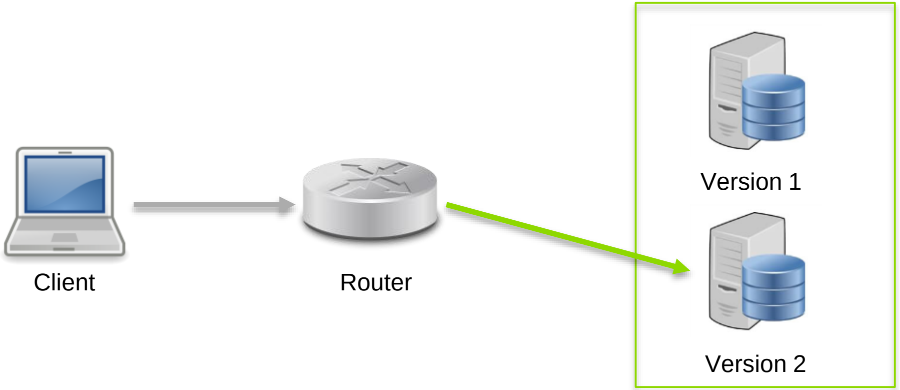
I nostri brand

LIBERO V:RGILIO PgCasa.it SUPEREVA DiLei SìViaggia QuiFinanza Buonissimo Pagine Gialle Pagine Bianche Tutto Città

Blue-green releasing



Blue version



Green version

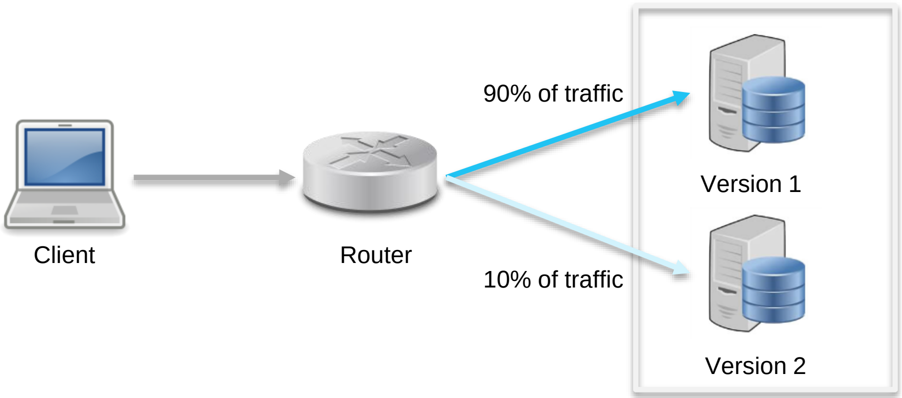
Blue-green releasing – Pros and Cons

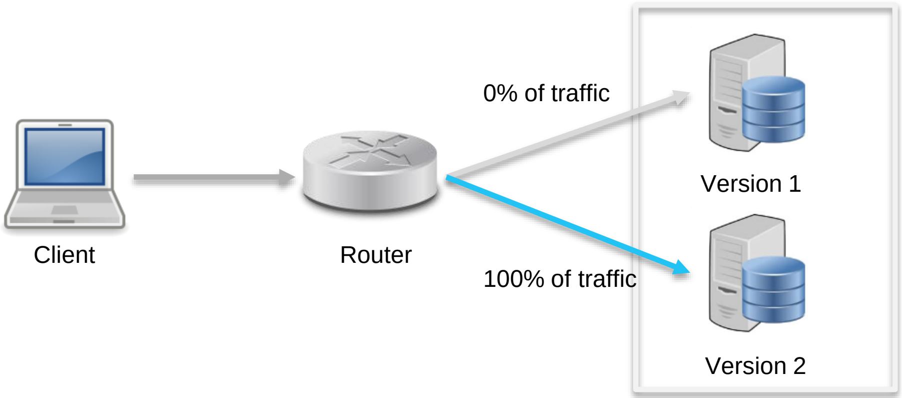
- Deploy a new version in production environment without downtime;
- Rapid switch from one version to other;
- Rapid rollback to the previous version;

but…

- All traffic is instantly direct toward the new release when the router switches;
- First time the whole infrastructure (network, remote resource, etc..) is tested with the new solution. Something could crash despite the code quality.

Canary releasing (dark launches)





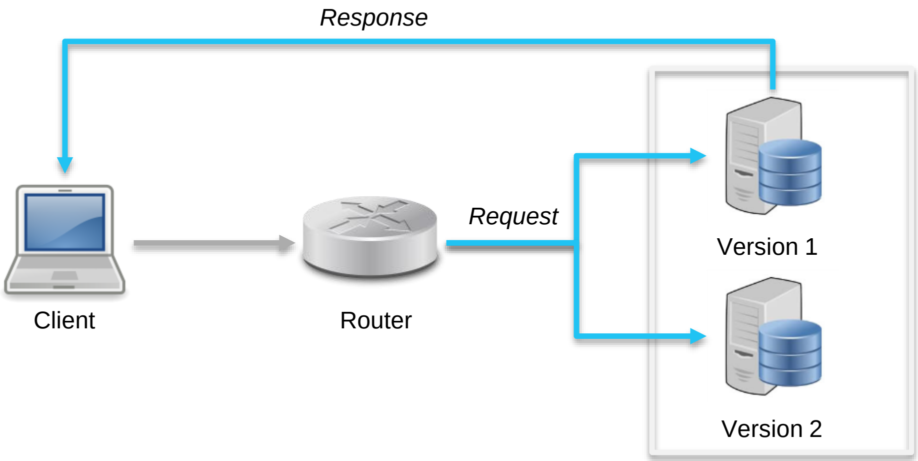
Canary releasing (dark launches) – Pros and Cons

- The same pros of Blue-green releasing;
- Useful for A/B testing;
- Low traffic to the new version which means easier monitor of the resources (network, CPU, database, etc..);

but…

- Longer deployment time;
- New releases blocked while not all traffic is switched to the new one; only bug fixing available.

Shadowing Traffic releasing



- Request sent to both version; only the current release provides the Response.

Shadowing Traffic releasing – Pros and Cons

- Real traffic to the new version without side effects due sudden application failure;
- Turn on/off the new release without knowledge of the client.

but…

- Longer deployment time;
- New releases blocked in this time; only bug fixing available;
- Duplication of traffic from the shared resources (dabatase, …).

italiaonline

Reference

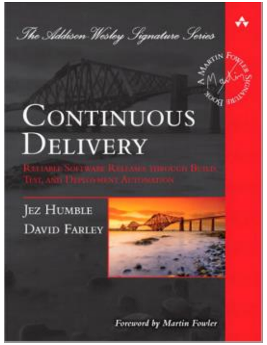
Tutto Città

Tutte le strade portano qui

I nostri brand

LIBERO VIRGILIO PgCasa.it SUPEREVA DiLei SìViaggia QuiFinanza Buonissimo Pagine Gialle Pagine Bianche Tutto Città

Reference



- Continuous Delivery by Jez Humble e David Farley

- Continuous Delivery with Jenkins + Docker + Git

  https://techannotation.wordpress.com/2018/11/28/continuous-delivery_jenkins_docker_git/

- Avoid stressful deploy of release!

  https://techannotation.wordpress.com/2019/03/20/avoid-stressful-deploy-of-release/

- Continuous Delivery Foundation

  https://cd.foundation/

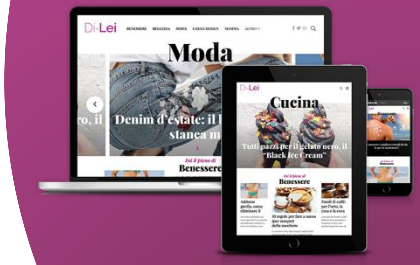
italiaonline

Any Question?

marco.ghisellini@Italiaonline.it

Di·Lei

Digitale, femminile, singolare



I nostri brand

LIBERO VIRGILIO PgCasa.it SUPEREVA Di·Lei SìViaggia QuiFinanza Buonissimo PagineGialle PagineBianche TuttoCittà

italiaonline

Il Nerd Happy Hour si prende una pausa, ci rivediamo il 10 Maggio con un nuovo interessante webinar.

Stay Tuned

I nostri brand

LIBERO VIRGILIO PgCasa.it SUPEREVA DiLei SìViaggia QuiFinanza Buonissimo Pagine Gialle Pagine Bianche Tutto Città