From Zero to CI/CD

Building your first CI Process

Nir Koren

DevOps CI/CD Team Leader, SAP Gigya Israel

My challenge / goals today

Each one of you will understand:

What is CI / CD

What is Git / GitHub and why most of the industry use it.

Introduction to build concepts

Jenkins CI

Each one of you will have:

“hello-world” CI process on his/ her Laptop.

What will we cover today?

- Introduction (s)
- Introduction to CI/CD
- **Introduction & hands-on** to Git and GitHub Basics
- **Introduction & hands-on** to Jenkins Basics
- Connect all together

What will we **NOT** cover today?

We are not going to develop anything today.

Not all of us comes from the same background and this not in the Workshop agenda.

Containerized applications: Docker / Kubernetes

Not enough time BUT it goes the same for those technologies as well.

Jenkins Pipelines and Shared libs

Not enough time but we will play with the pipeline in a nutshell.

Agenda

09:00 – 10:30: Introduction to CI / CD

10:30 – 11:00: Coffee Break 

11:00 – 12:30: Introduction to Git / GitHub

12:30 – 13:30: Lunch 

13:30 – 15:00: Introduction to Jenkins

15:00 – 15:30: Coffee Break 

15:30 – 17:00: Jenkins & connect all

WE WILL HAVE TO BE FLEXIBLE

$ whoami

Nir Koren, DevOps CI/CD Team Leader, SAP Gigya Israel
Doing and implementing DevOps and CI / CD for 10+ years


https://il.linkedin.com/in/nirkoren


https://www.facebook.com/koren.nir


@KorenNir


@nir_koren

I work for SAP Gigya

100% Cloud SaaS company

Leader of customer identity and access management

Part of SAP Customer Data Cloud solutions

200 / 100K Employees

Acquired by SAP in 2017

THE 2014 LEADERBOARD
OF JAVA TOOLS & TECHNOLOGIES

REBELLABS

82.5%
JUnit*
TOP TESTING FRAMEWORK USED BY DEVELOPERS

70%
Jenkins°
MOST USED CI SERVER IN THE INDUSTRY

64%
Maven
MOST USED BUILD TOOL IN JAVA

64%
Nexus°
THE MAIN REPOSITORY USED BY DEVELOPERS

69% Git*
#1 VERSION CONTROL TECHNOLOGY OUT THERE

67.5%
Hibernate*/°
THE TOP ORM FRAMEWORK USED

65% Java 7
THE INDUSTRY LEADER FOR SE DEVELOPMENT

56%
MongoDB*
THE NOSQL TECHNOLOGY OF CHOICE

55%
FindBugs*/*
MOST-USED STATIC CODE ANALYSIS TOOL

50%
Tomcat*
THE MOST POPULAR APPLICATION SERVER

49%
Java EE 6*
FOUND IN THE MOST ENTERPRISES

48%
Eclipse
THE IDE USED MORE THAN ANY OTHER

40% **Spring MVC*/***
MOST COMMONLY USED WEB FRAMEWORK

32% **MySQL***
THE MOST POPULAR SQL TECHNOLOGY

* *Multiple selections possible*

° *Normalized to exclude non-user base*

Sample population of 2164 Java professionals, sample error 2.1%

RebelLabs Tools and Technologies Leaderboard 2016

REBELLABS by ZEROTURNAROUND

Maven

Over **two in three (68%) devs** useMaven as their main build tool

git

Over **two in three (68%) devs** use Git as their version control

Java 8

Almost **two in three (62%) devs** use Java 8 in production

Jenkins

Three in five (60%) devs use Jenkins for CI

IntelliJ IDEA

Almost **one in two (46%) devs** useIntellij, the most popular IDE in the survey.

spring

Over **two in five (43%) devs** use Spring MVC

Tomcat

Over **two in five (42%) devs** useTomcat server in production

ORACLE DATABASE

Almost **two in five (39%) devs** use Oracle DB in production

Microservices

Over **one in three (34%) devs** have adopted a microservices architecture

docker

Almost **one in three (32%) devs** use Docker in production

Java EE 7

Over **three in ten (31%) devs** use Java EE 7

spring

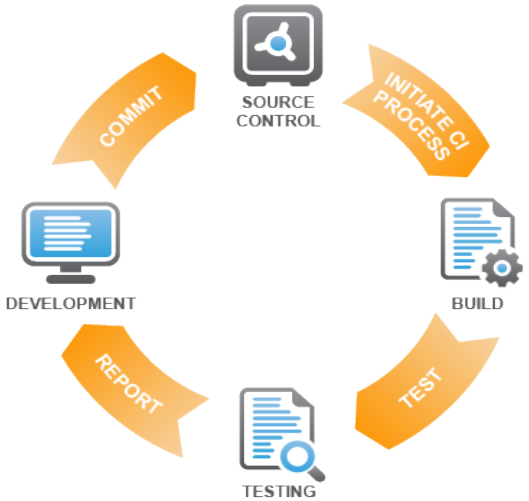
Almost **three in ten (29%) devs** use Spring Boot

Introduction to

CONTINUOUS INTEGRATION / DELIVERY

What is Continuous Integration?

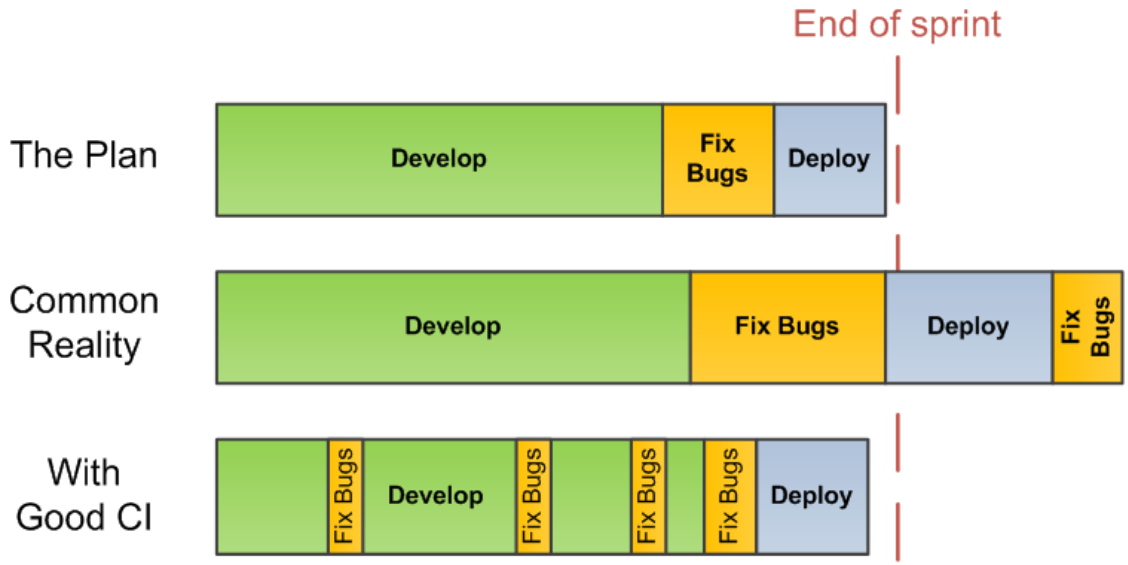
"A key **software development practice** where members of a team **integrate their work frequently**."



Martin Fowler, Continuous Integration:
http://martinfowler.com/articles/continuousIntegration.html

What is Continuous Integration?

Continuous integration involves integrating early and often, in order to avoid "integration hell“



CI / CD Main Goal

Provides a **rapid feedback**.
If a defect is introduced into the code base, it can be **identified** and **corrected ASAP**.

Why CI/CD?

- Frequent changes ➔ Less integration problems.
- Bugs are detected earlier ➔ Saves money.
- Avoid last minute chaos.
- Transparency to all.
- Testing on Production-Like env.
- Easy to rollback in case of any issue.
- Enforce of automation culture.
- Enforce DevOps culture.
- Make the developers accountable and take ownership.

CI Principles

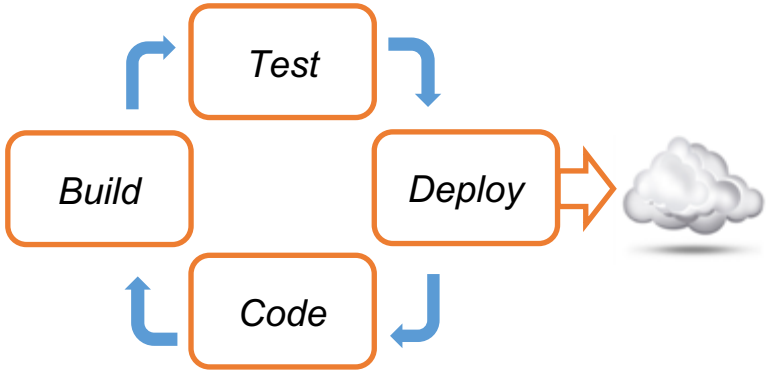
- **Automate** the build (single command), **Automate all**.
- Build is **self testable**.
- Baseline **branch is open consistently**.
- **Every commit** should be built.
- Build should be **fast**.
- Test Env is **clone of Production** (as much as we can).
- Everyone can see the status – **Transparency**.

What is

CONTINUOUS DELIVERY / DEPLOYMENT

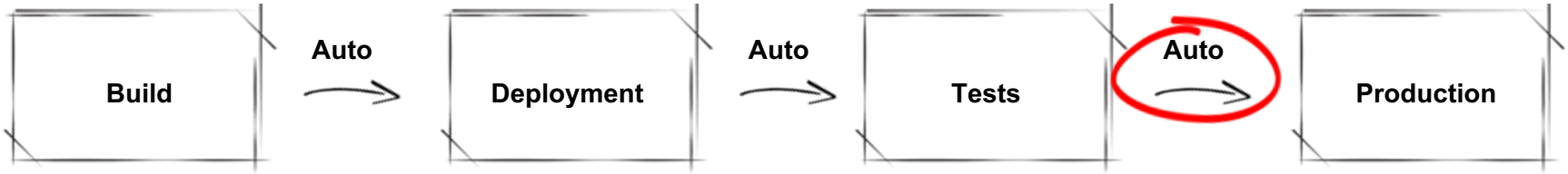
Continuous Delivery & Deployment

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

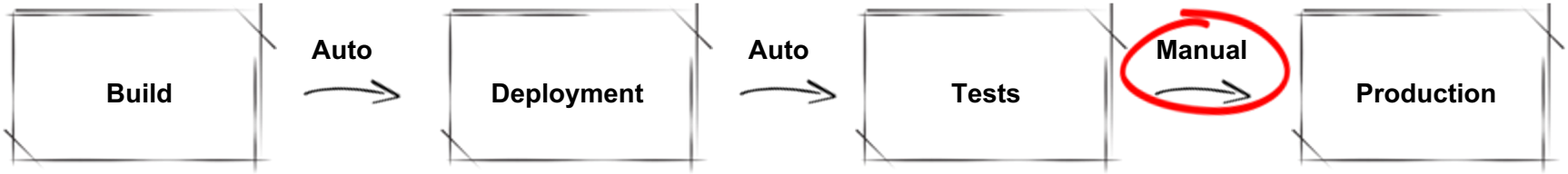


Continuous Deployment means that every change goes through the pipeline and automatically gets put into production, resulting in many production deployments every day.

Continuous **Deployment**



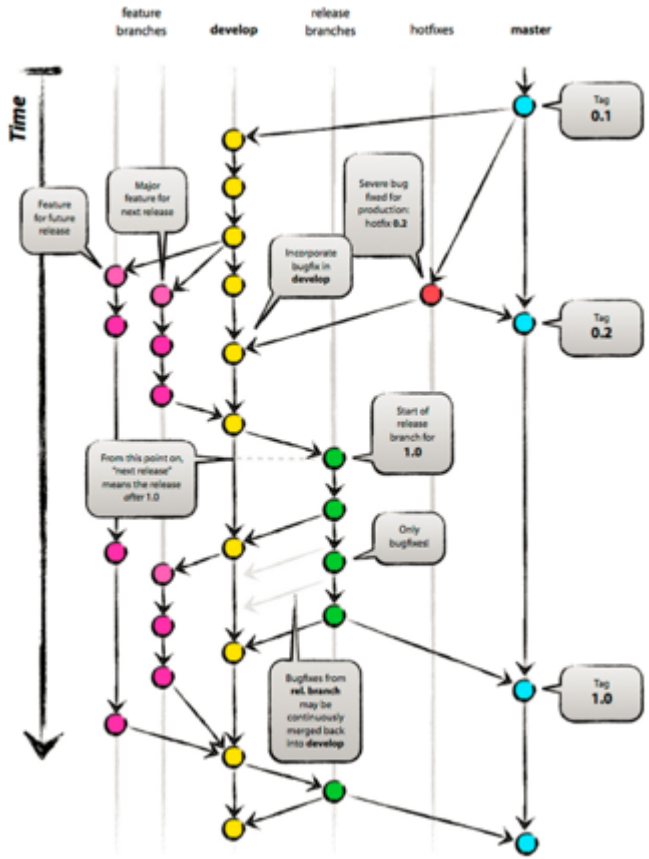
Continuous **Delivery**

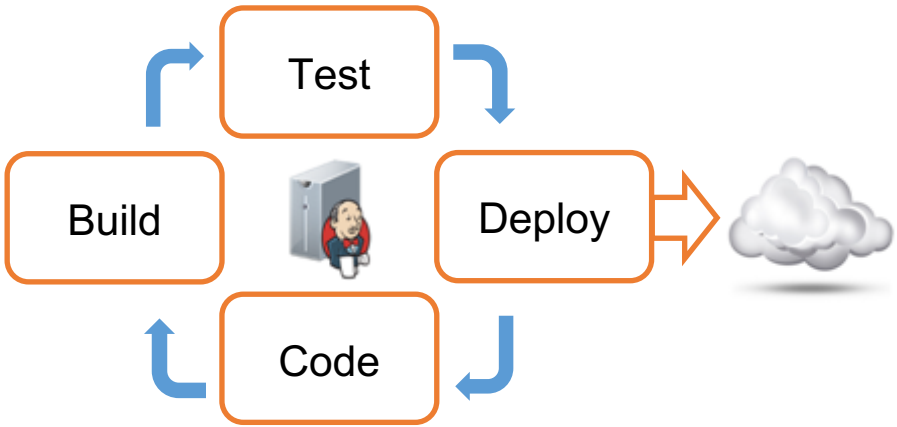


CD Principles

- The Software is always **deployable** through it's lifecycle.
- Anybody can get **fast and automated state** of Production.
- You can perform **push-button deployments** any time

CD Common structure



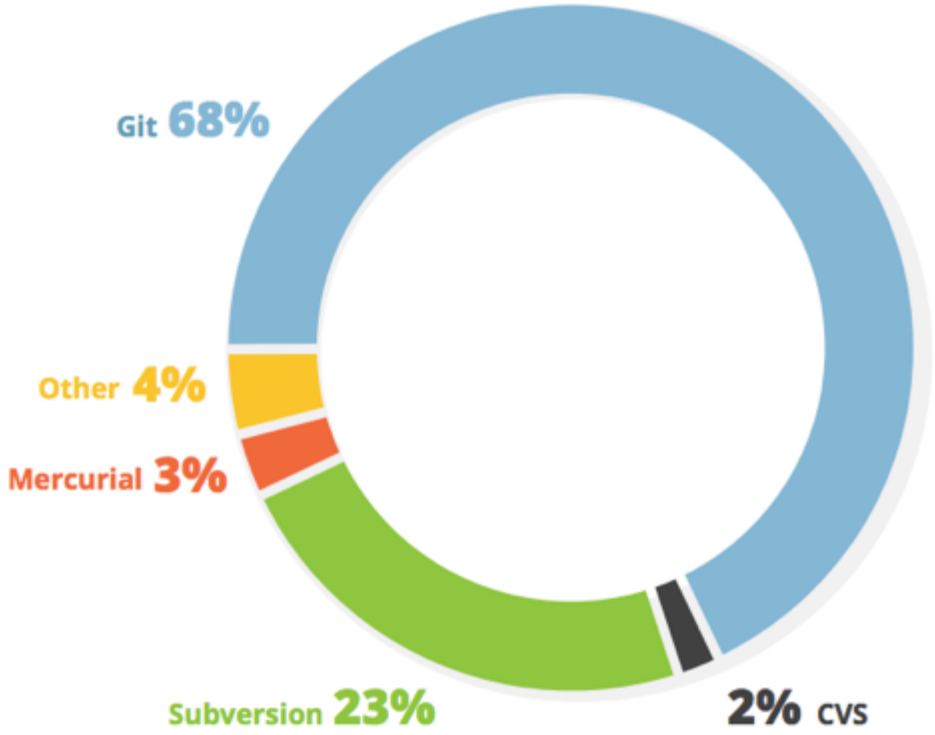


CD is not only in Computer Software

Tesla Model S gets firmware updates on regular basis for both UI and major elements (suspension, acceleration and more)

Introduction to
Git & GitHub

Figure 1.18 Most Commonly Used VCS



Lets install **git** & create **GitHub** account

Linux based: $ sudo apt-get install git

Windows: Download from https://git-scm.com/

Mac: $ brew install git Or download from https://git-scm.com/

Initial configurations:

$ git config --global user.name "John Doe"
$ git config --global user.email johndoe@example.com

Check your configurations:

$ git config -l

https://github.com/

Sign Up

Create personal access token
https://github.com/settings/tokens

Git installation / configuration files.

INSTALLATION DIRECTORY

CONFIGURATION DIRECTORY

CODE REPOSITORY


```
[user]
        name = Nir Koren
        email = nirk@liveperson.com

[merge]
  tool = p4merge
…
```

What is Git?

Git is an **Open Source**, **Distributed** and **popular** version control system.
Designed for **Speed** and **Efficiency**

Open Source

- Free
- Popular
- Highly contributed by the community and GitHub

Popular

- Multi-platform support
- Multiple GUI’s
- Fully command lined – great for CI / Automation
- Contains any IDE / Integrated plugins

Fast

No Network needed.

- Performing a diff
- Viewing file history
- Committing changes
- Merging branches
- Obtaining any other revision of a file
- Switching branches

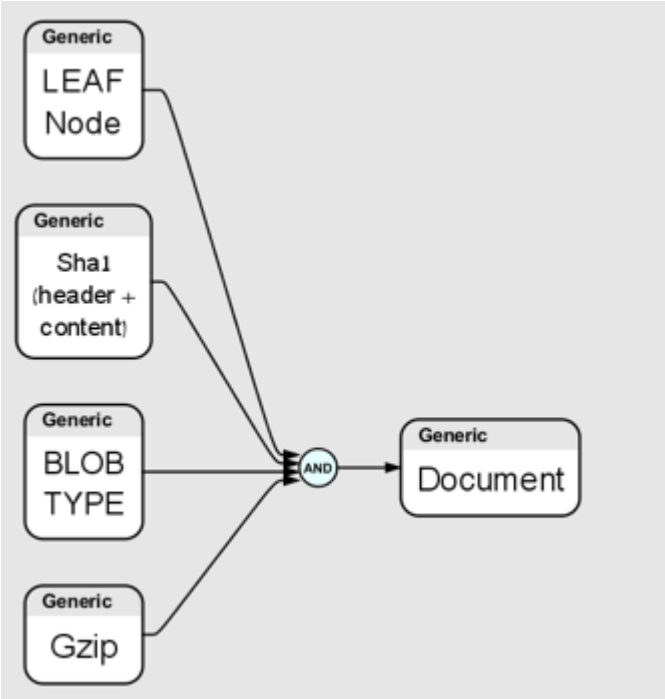
History & Founder

First release @ 2005 for Linux Kernel source code.
New free alternative for BitKeeper.

I'm an egotistical bastard, and I name all my projects after myself. First “Linux” and now “git”.

Linus Torvalds

Git: Under the hood



Key Concepts

COMMIT

A **Snapshot** in the repository
State of the project in a certain time. Point of a tree

BRANCH

A label with **user-friendly name**. Point to a **commit**.

TAG

A descriptive name to a **list of commits**.

Technically – it’s all the same

Key Concepts

REMOTE

Can be **origin** (where it cloned from) or **upstream** (custom)
The version that hosted in the server. Can be **remote host** or **local clone**.

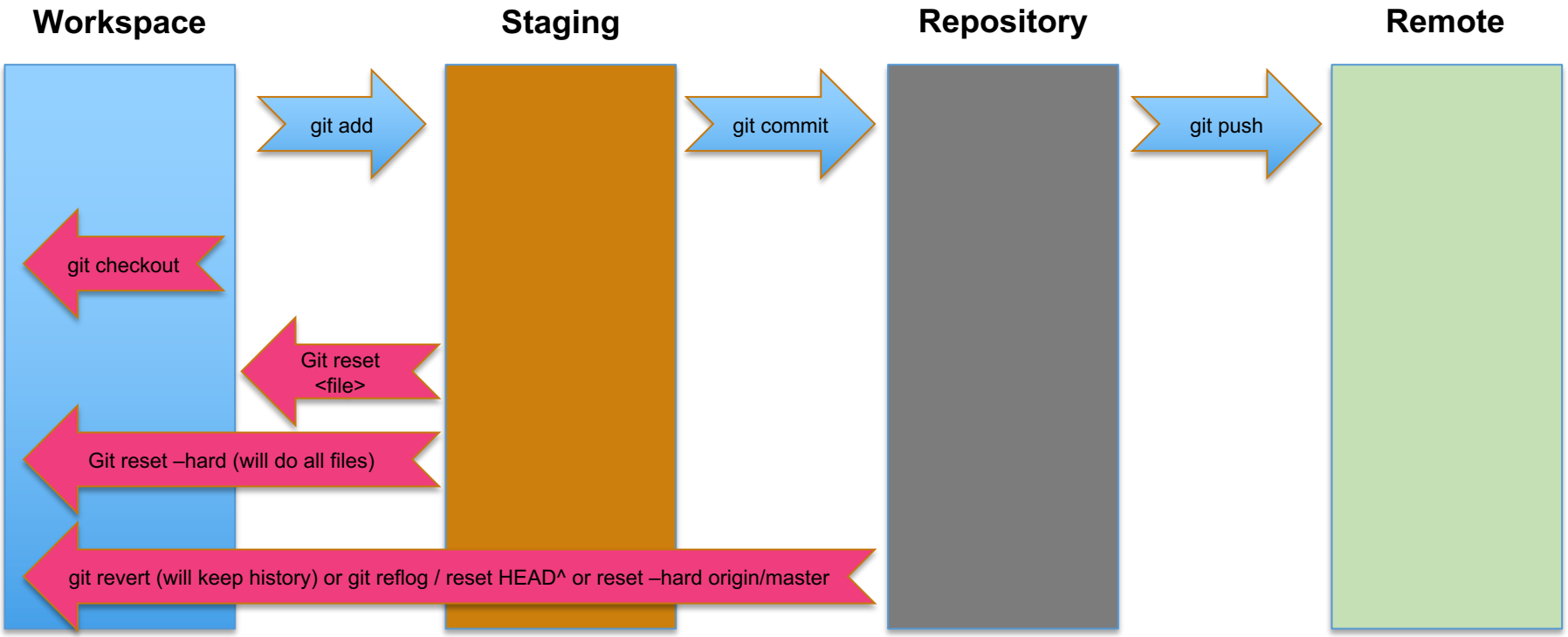
CLONE

Copy of a repository locally on your laptop. Comes with metadata relationship to the remote repo

FORK

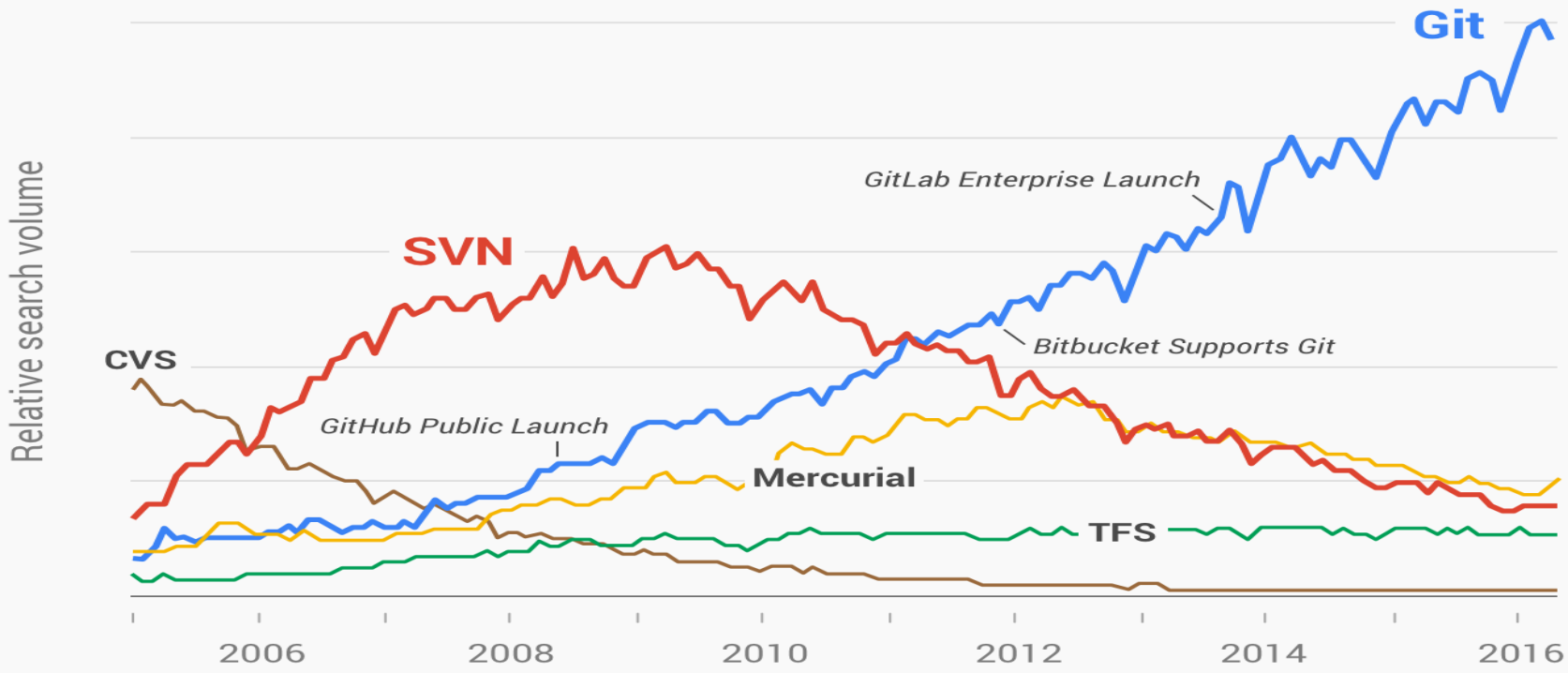
A **personal copy** of a repository that lives **remotely** on a user account and holds relationship with the original repo..

Git: Areas


Workspace
Staging
Repository
Remote
git add
git commit
git push
git checkout
Git reset
<file>
Git reset –hard (will do all files)
git revert (will keep history) or git reflog / reset HEAD^ or reset –hard origin/master

How Did SVN Manage to Lose?

Version control interest over time



GitPrime

Data: Google Trends, GitHub, Atlassian, GitLab

Companies & Projects Using Git

Google facebook Microsoft twitter LinkedIn NETFLIX PostgreSQL

android RAILS Qt GNOME eclipse

Perforce new products contains integration to Git

The best way to learn Git

FORGET Perforce
FORGET ClearCase
FORGET SVN
FORGET Any VCS you know

It’s much easier to teach a person who doesn’t know any VCS

Philosophy

- While **SVN trunk** is development, **GIT Master** is Release.
- Anything in **origin master is deployable**.
- Create a descriptively **named branch** for new feature development
- **Commit early and often** to new branch locally
- Ensure that **each commit represents one idea** or complete change.
- Push your new branch work to the **same named branch on origin**

Introduction to

What is GitHub?

Web-based hosting service for version control using Git.
The largest source code in the world

GitHub Company

- Founded in 2007. Owned by Microsoft
- Headquartered in San Francisco, 2500+ employees
- GitHub.com > 73M users, 200M Repositories
- Core GitHub developers are core Git OS contributors
- Cool brand (shop, merchandizing, crazy company)

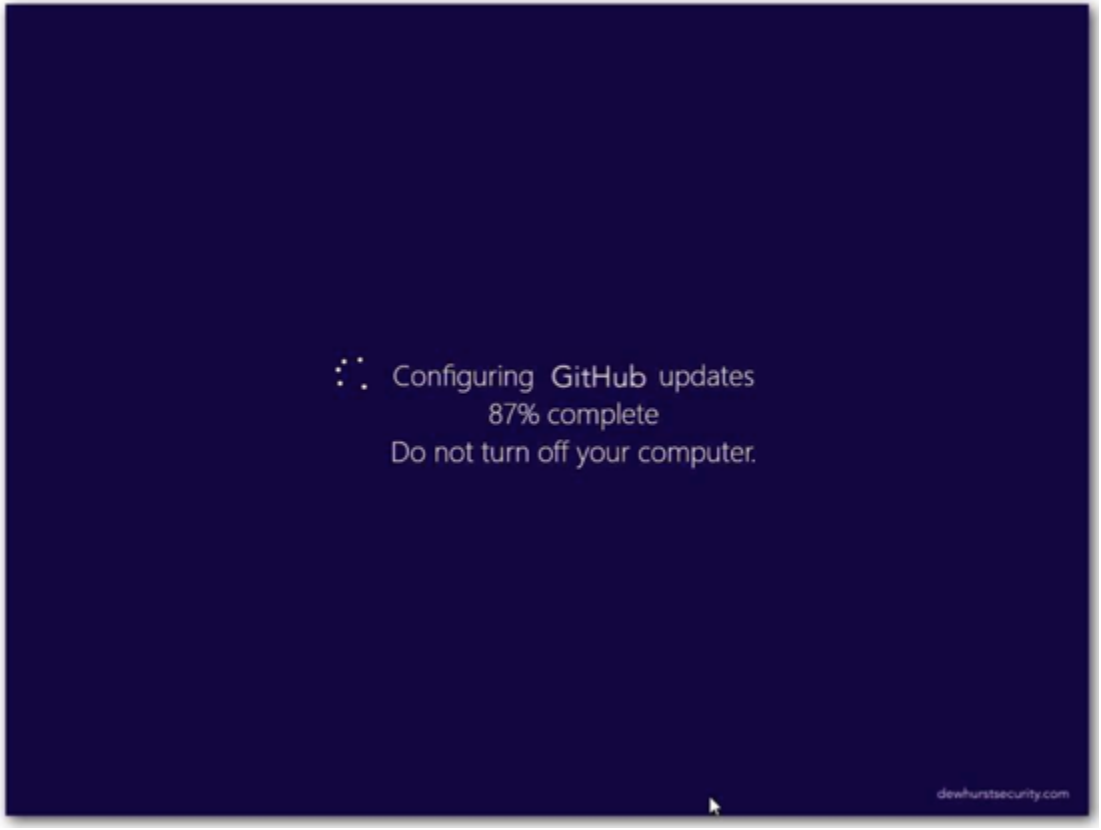
Any Microsoft employees here?

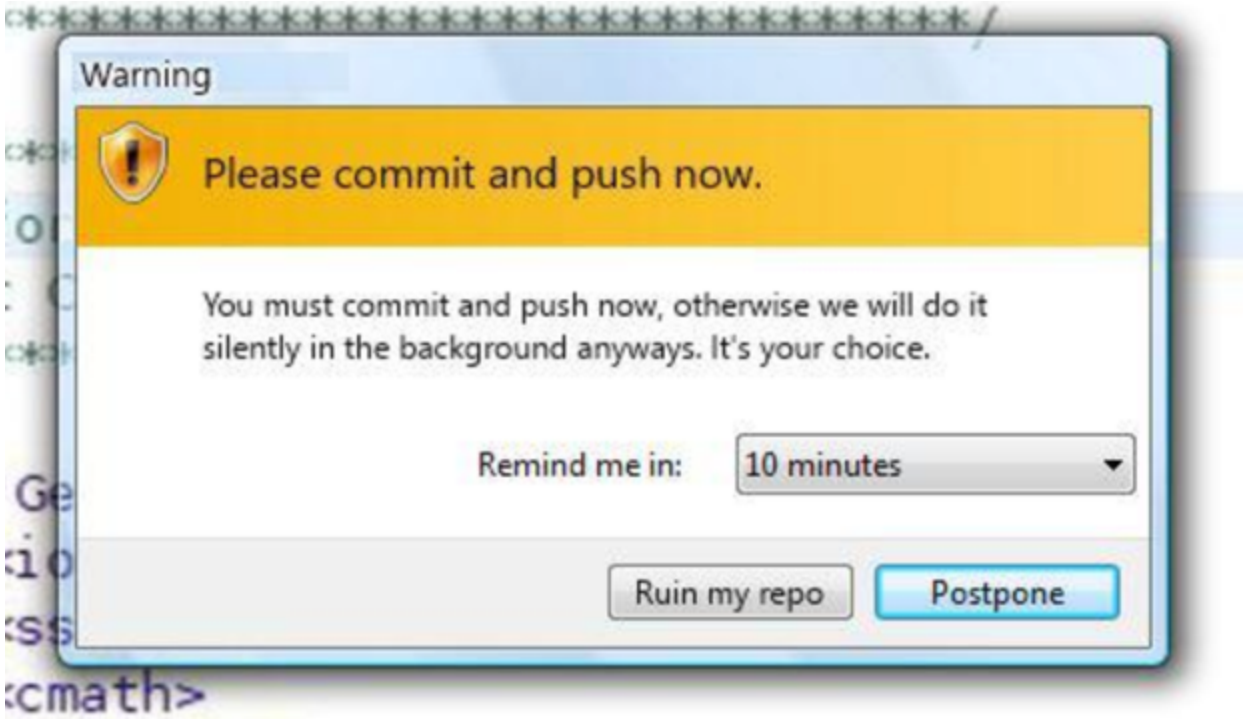
Microsoft

GitHub 2019

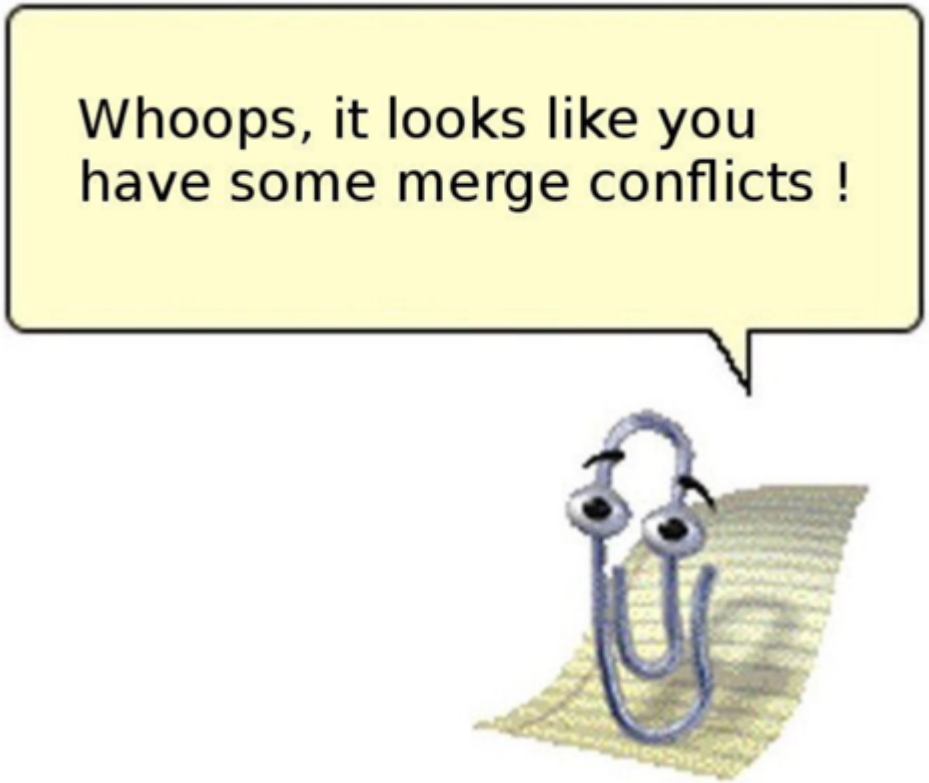
Home edition

Please enter your product key :


Configuring GitHub updates
87% complete
Do not turn off your computer.
dewhurstsecurity.com


Warning
Please commit and push now.
You must commit and push now, otherwise we will do it silently in the background anyways. It's your choice.
Remind me in:
10 minutes
Ruin my repo
Postpone

- GitHub Starter
- GitHub Home Basic
- GitHub Home Premium
- GitHub Professional
- GitHub Enterprise
- GitHub Ultimate


Whoops, it looks like you
have some merge conflicts !

Just Kidding ☺

Pricing models

- **Free:** contains part of the features used.
- **Team:** extra features and privacy.
- **Enterprise (hosted):** private in the cloud Github.com
- **Enterprise (Enterprise):** On-premise. Install and maintain on your own.

Partial features list of

WHY GITHUB?

In-place editing using Ace Editor

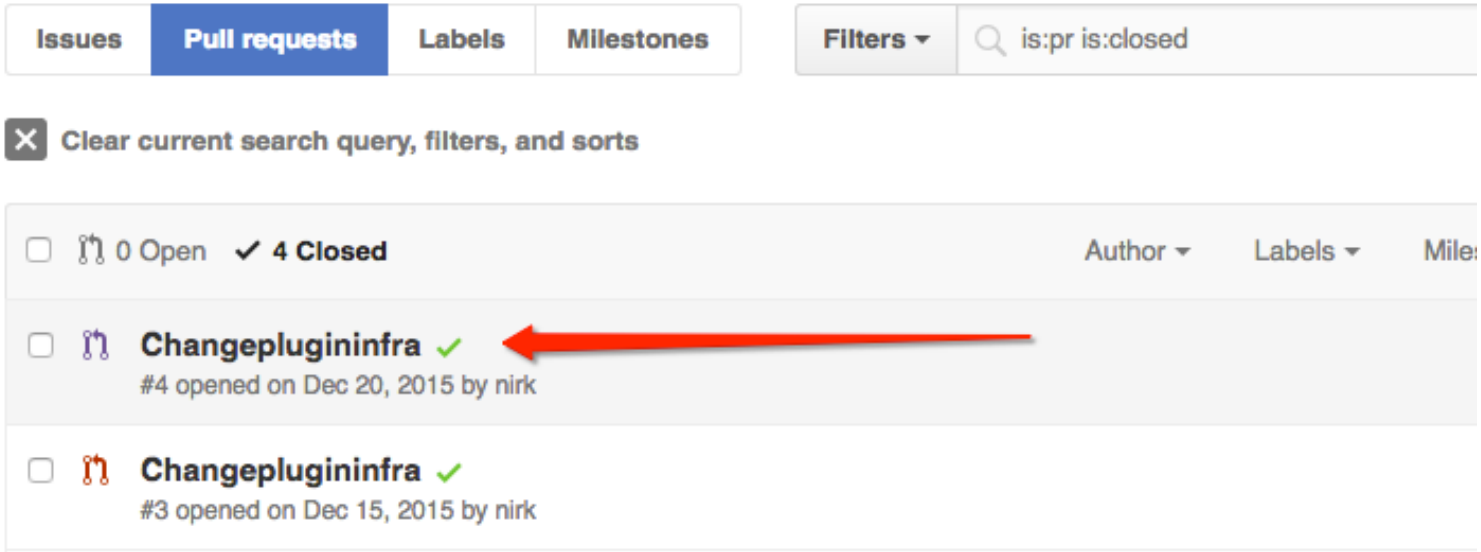
Nir Koren init commit — Latest commit dc3e34a on Nov 24, 2021 — **History**

0 contributors

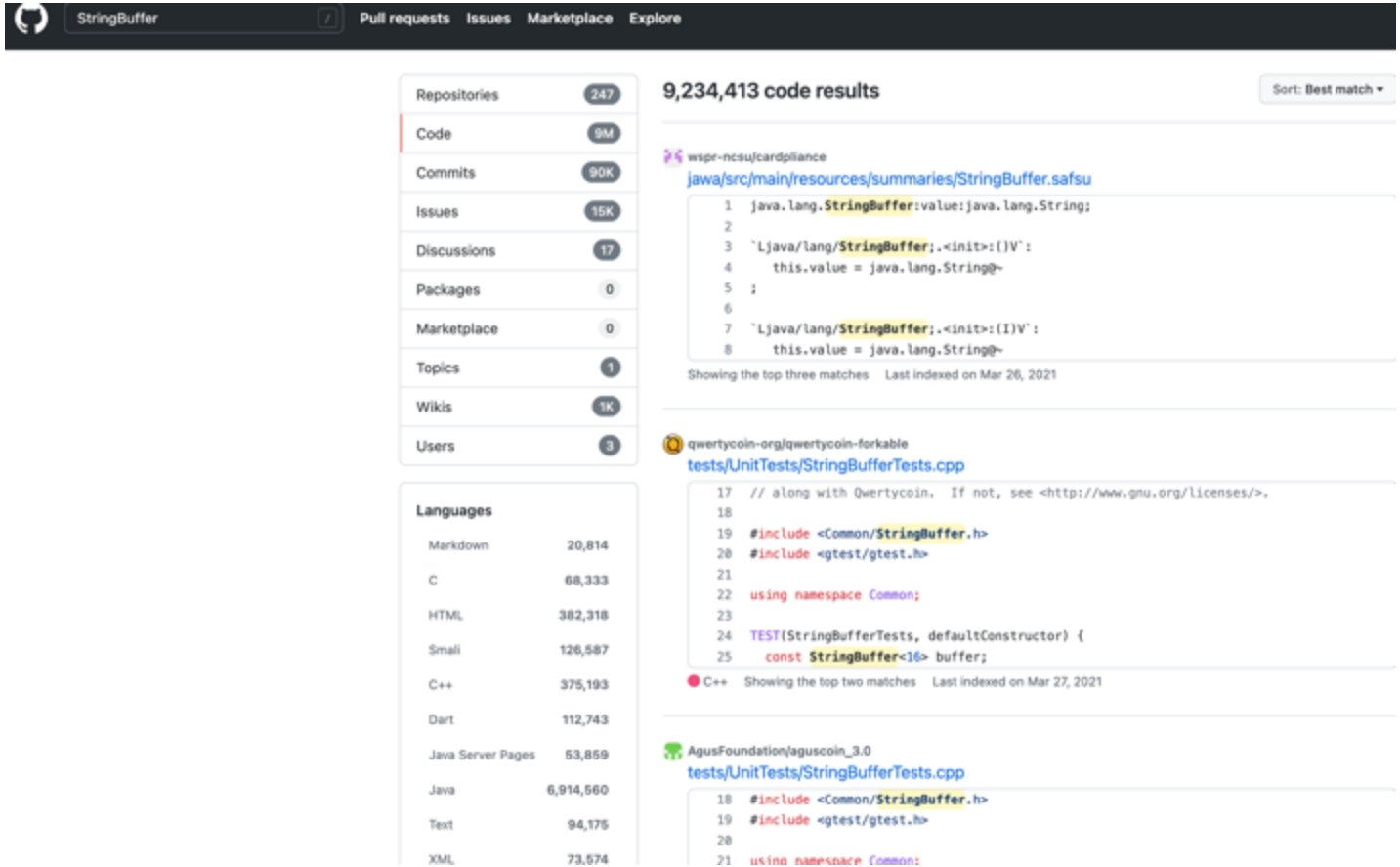
17 lines (13 sloc) | 406 Bytes — Raw | Blame

```
1  package org.nirkoren.maven.demo;
2
3  import org.junit.Assert;
4  import org.junit.Test;
5  import org.nirkoren.maven.demo.HelloHandler;
6
7  public class HelloHandlerTest {
8
9          @Test
10         public void validateNameNotNull() {
11                 HelloHandler handler = new HelloHandler();
12                 String response = handler.sayHello();
13                 // Due to the code: This test will never fail :)
14                 Assert.assertNotNull("String got null value",response);
15         }
16
17 }
```

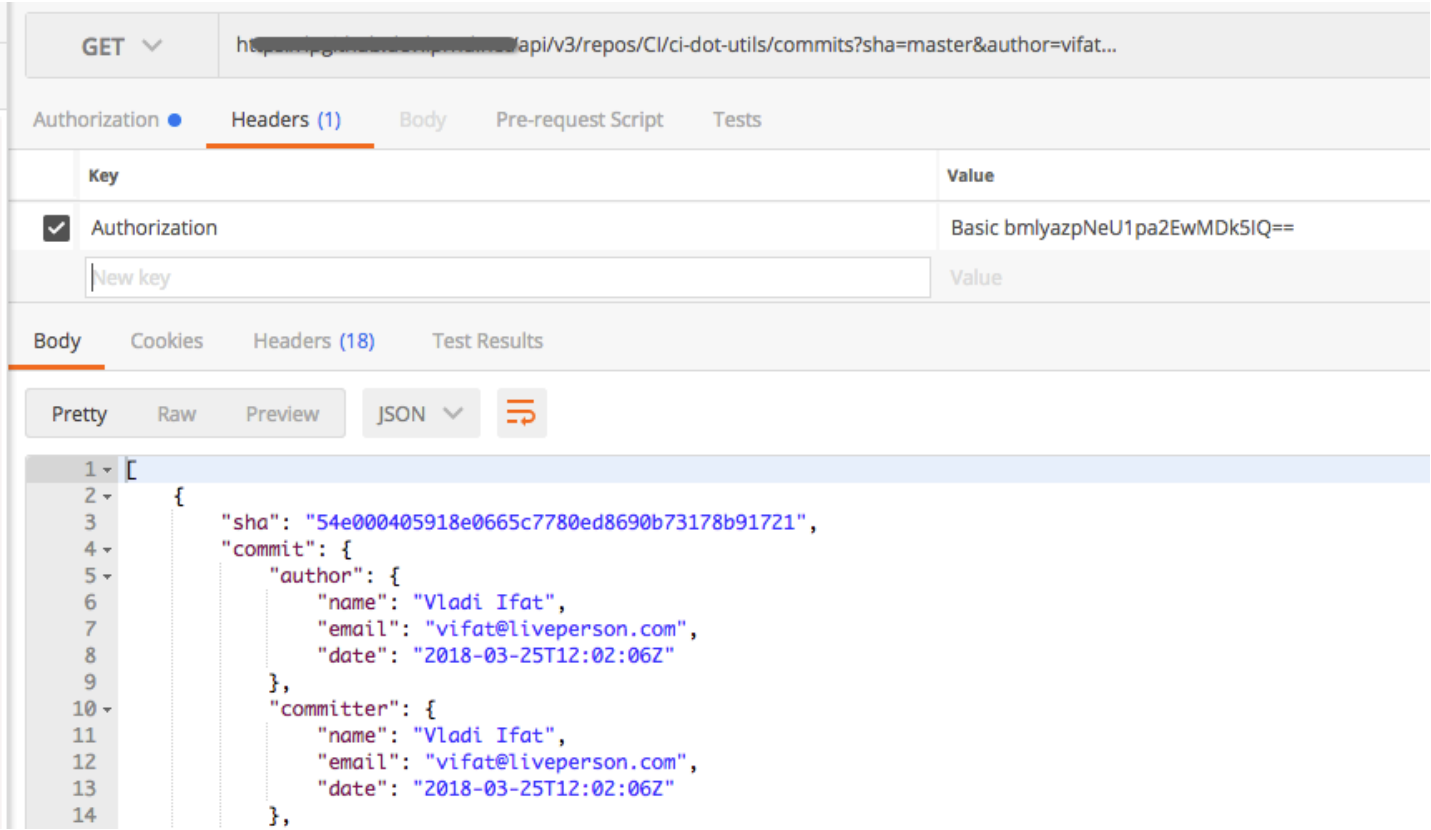
Both Machine and Human Code Review



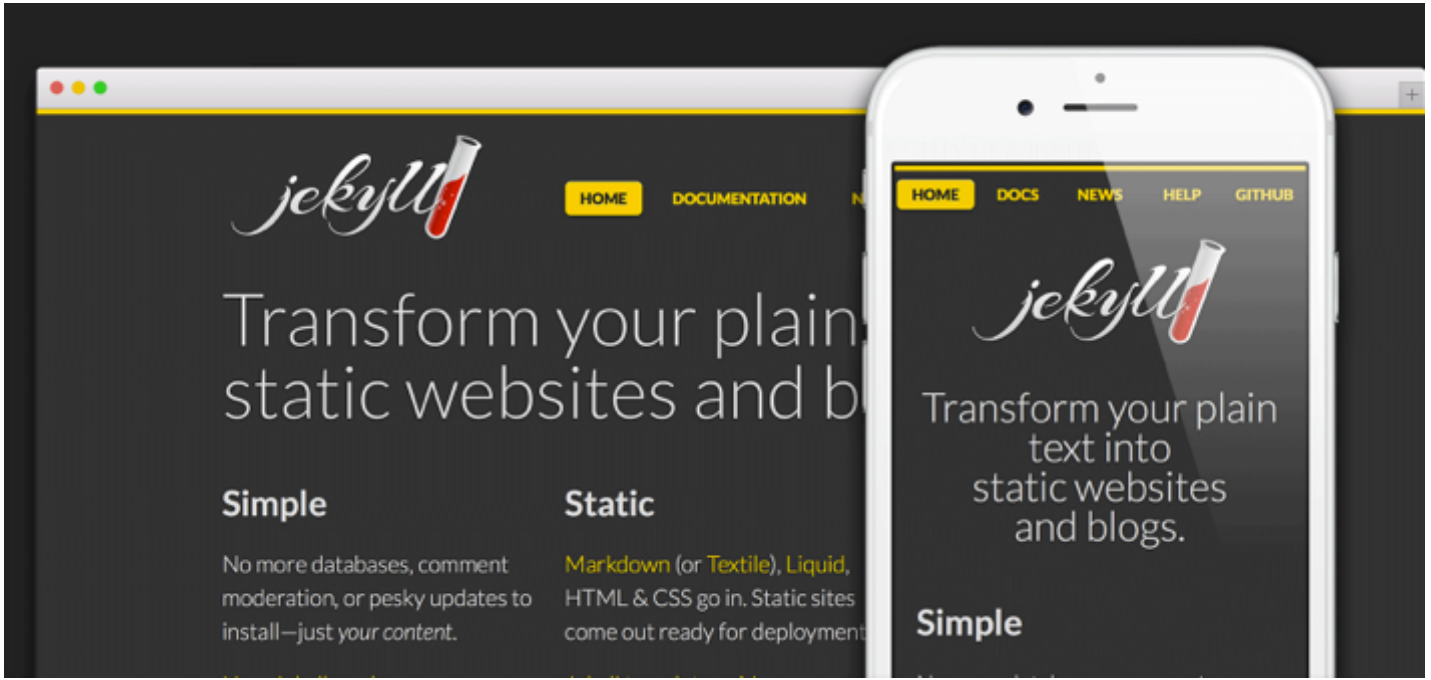
Code Search engine



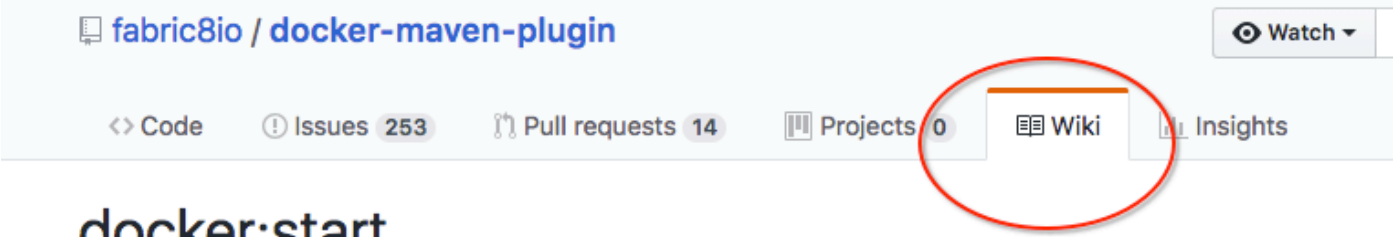
Rich documentation and API



GitHub Pages



GitHub WIKI



docker:start

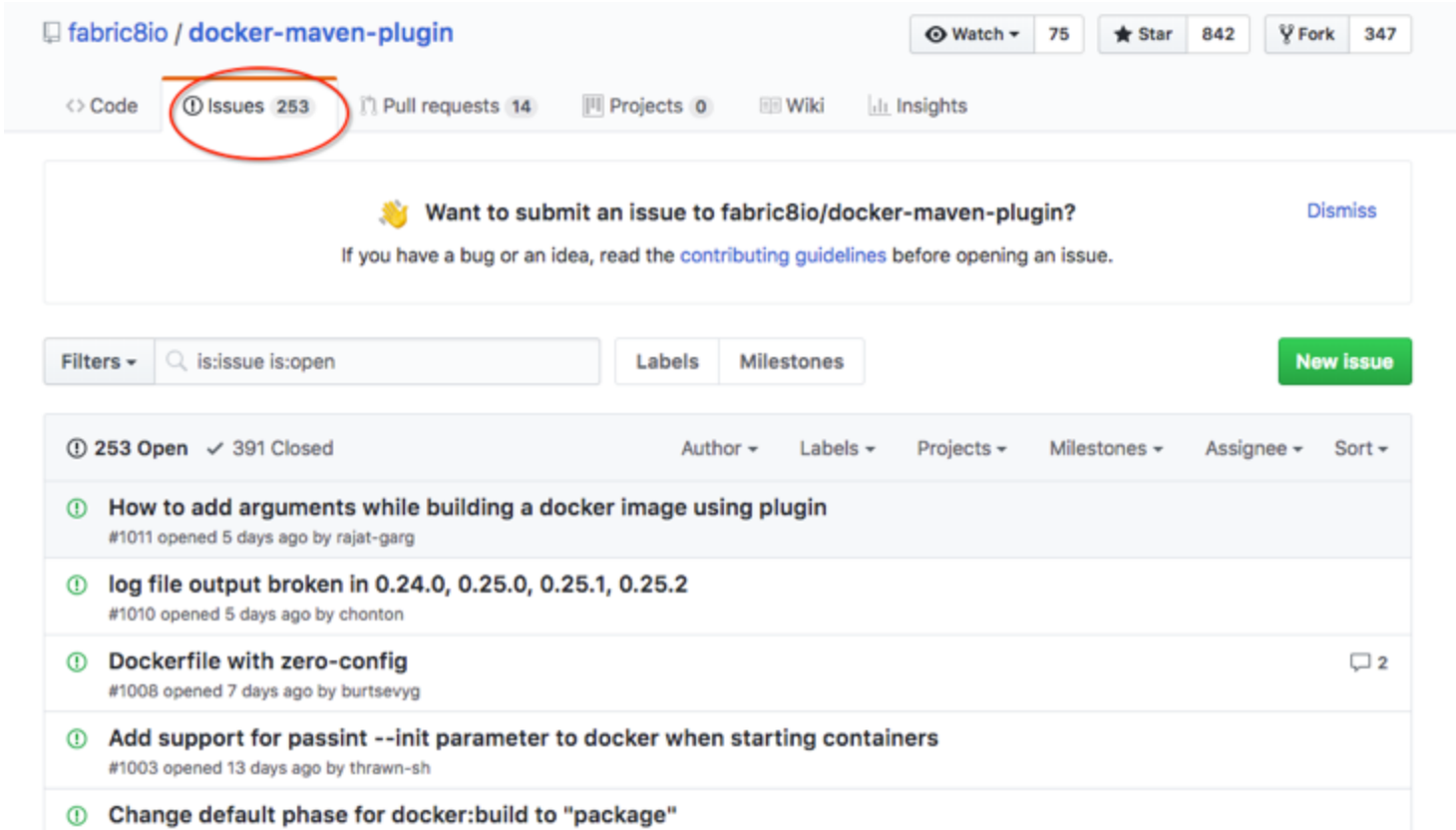
Roland Huß edited this page on Feb 27, 2015 · 3 revisions

docker:start

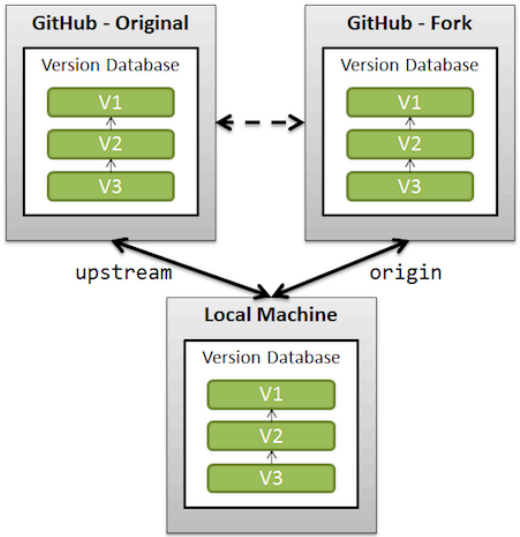
Maven Goal for starting a container. This goal has various configuration parameters which influence its behaviour. This goal is best attached to the `pre-integration-test` phase of the Maven lifecycle, so that containers are started before the integration test runs.

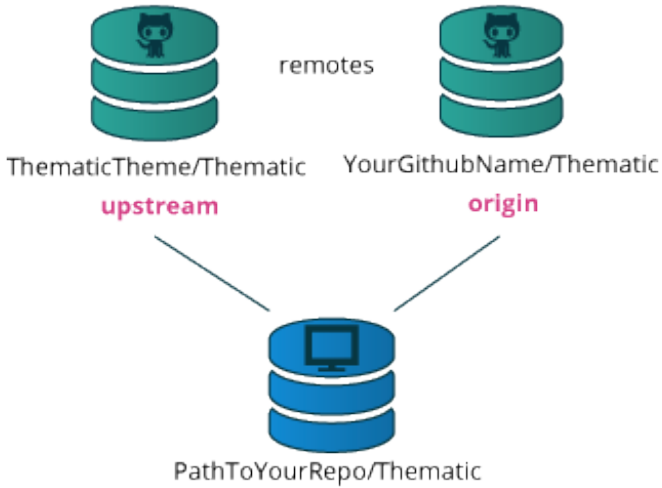
If multiple containers are required they should be attached individually to the lifecycle, each with their own configuration. Common configuration options then should go into the main configuration section. Creates and starts a docker container.

GitHub Issues



Forks model – Open Source work model

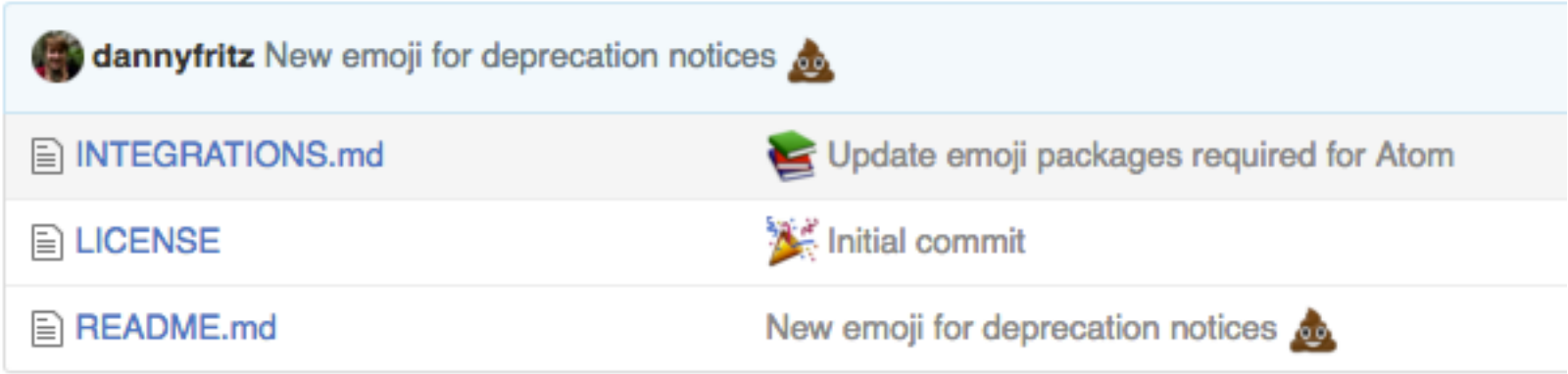




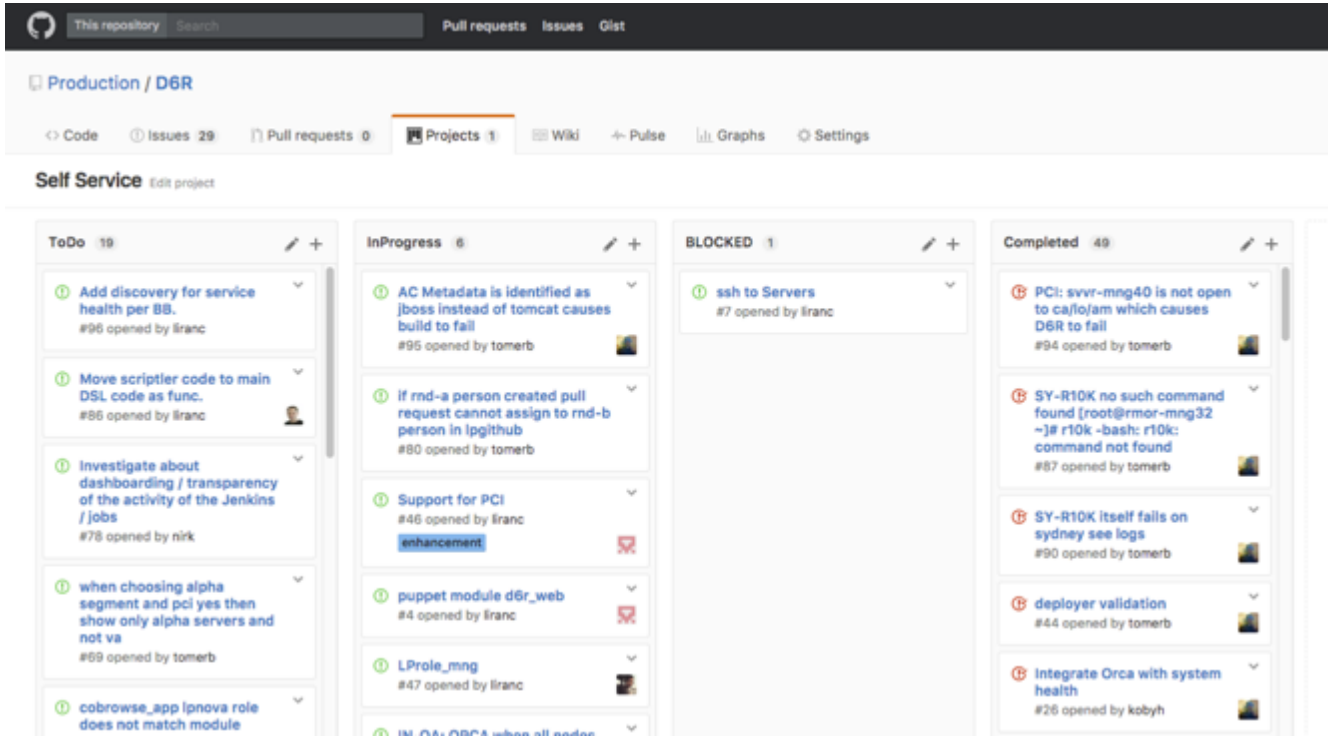
Pull Request

Tell others about your change, Let human / machine to review your code.

Communicate with Emojis



Manage your tasks in Board



Key Concepts

ORGANIZATION

Group of **2 or more users** that typically acts like a real org / company. It administrated and contains teams, users and repos.

REPOSITORY

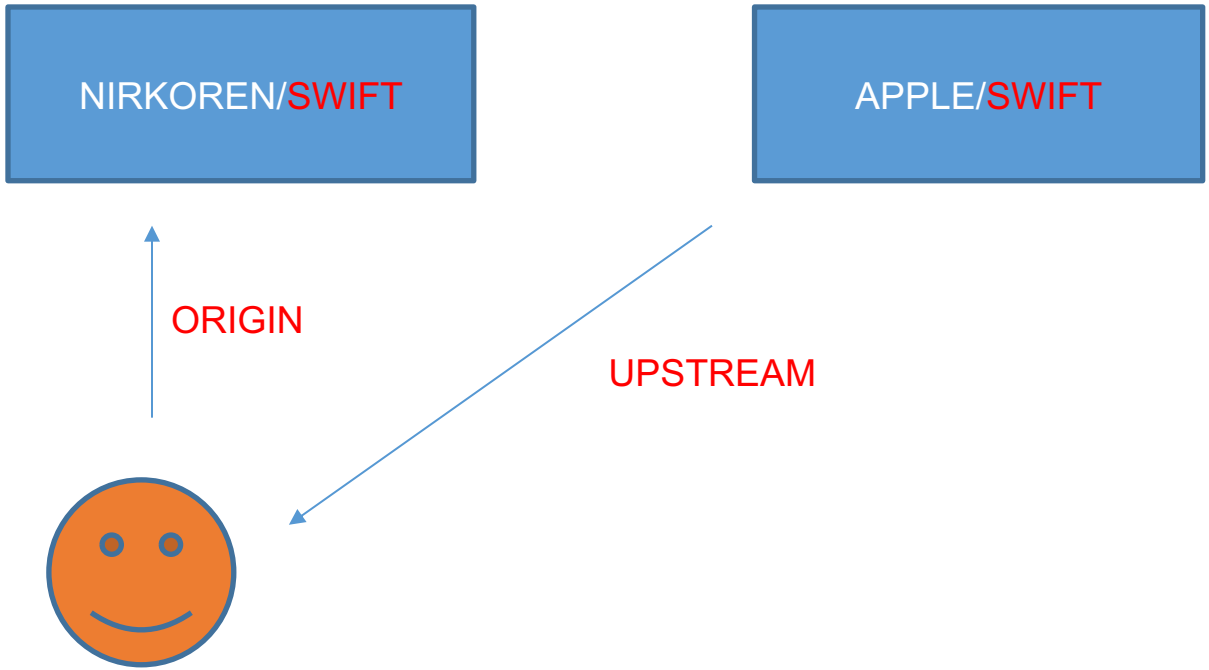
Collection of files with metadata (.git folder) that contains history, revisions etc’

COLLABORATOR

A person who had read / write permissions to a repository.

CONTRIBUTOR

A person who contributes to a repository but has no collaborator access. Only PULL requests.

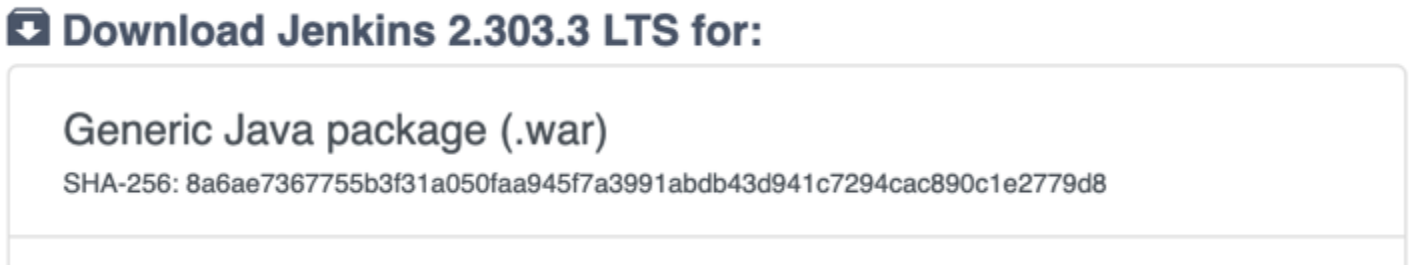

NIRKOREN/SWIFT
APPLE/SWIFT
ORIGIN
UPSTREAM

Practice

Network might be slow

In the meantime - Please download from

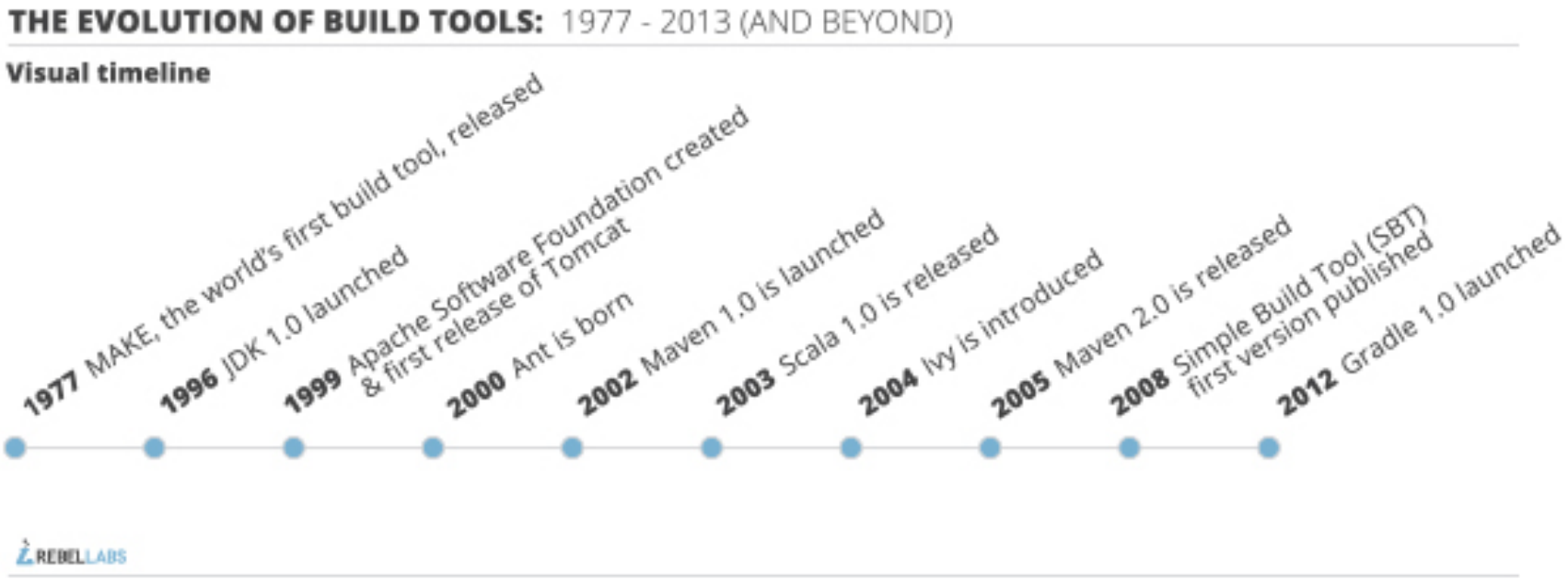
https://jenkins.io/download/



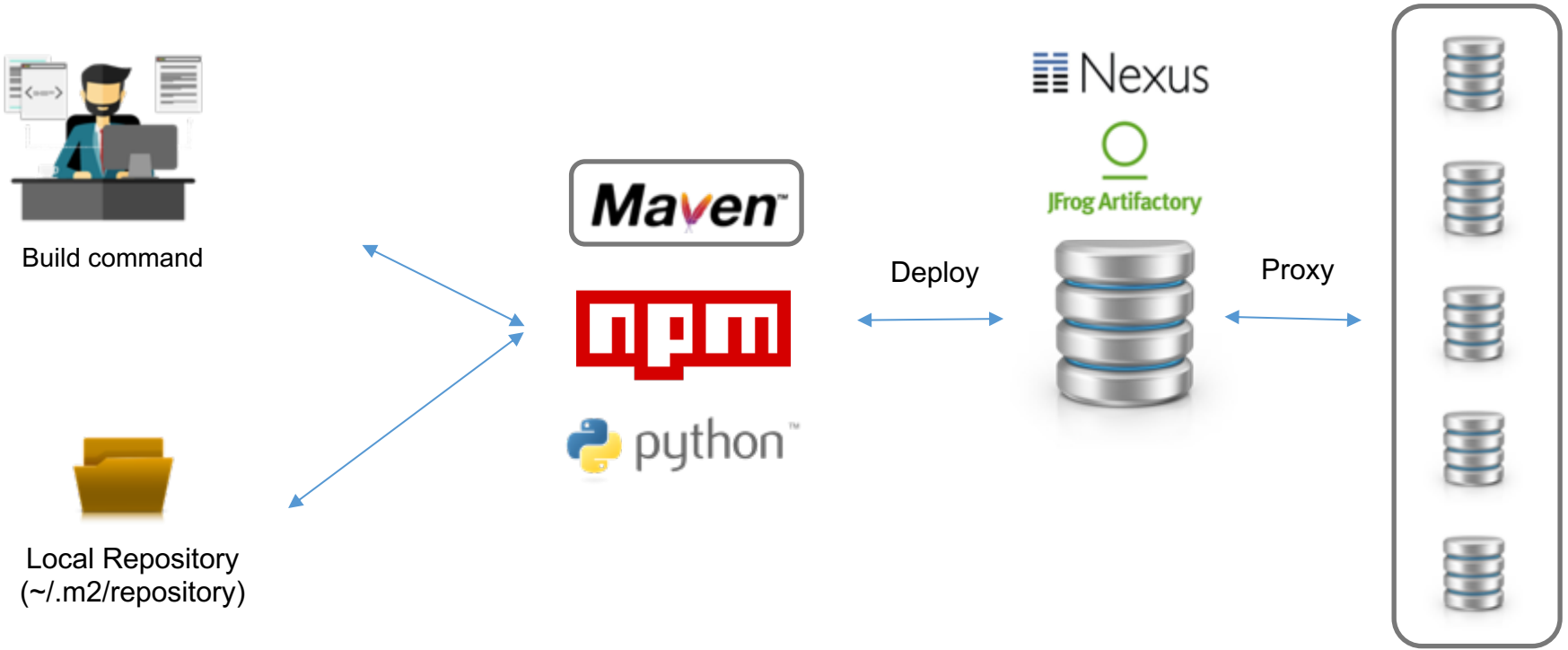
Introduction to
Build processes

What is “Build”?

History of build tools



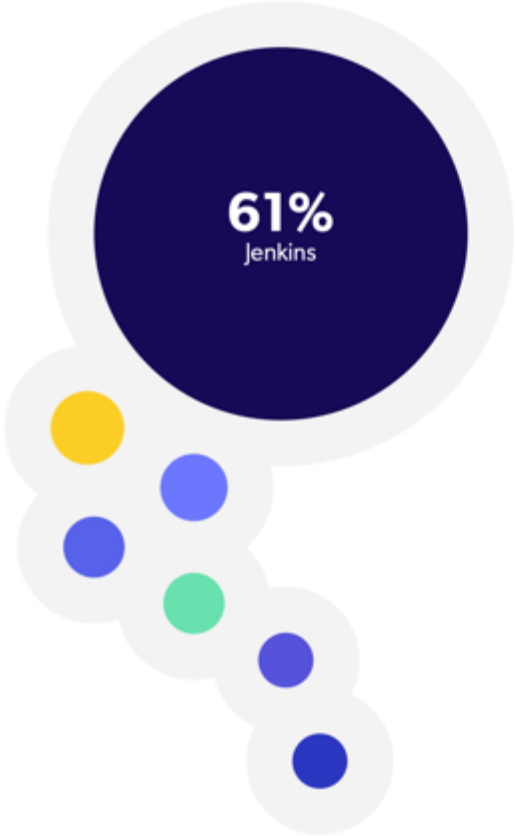
Common Architecture



Common build pipeline

Introduction to
Jenkins CI Server

Which CI/CD technologies are you using?



Jenkins **61%**

Bamboo **12%**

Travis CI **11%**

N/A **10%**

Other **10%**

TeamCity **9%**

Circle CI **9%**

Let’s install Jenkins first

- Download Jenkins (Preferred LTS) https://jenkins.io/
- Open up a terminal in the download directory.
- Run java -jar jenkins.war --httpPort=9090.
- Browse to http://localhost:9090.
- Follow the instructions to complete the installation.

Docker?

$ docker run -d -p 9090:8080 -u root -v /Users/nirk/jenkins_home:/var/jenkins_home jenkins/jenkins:lts

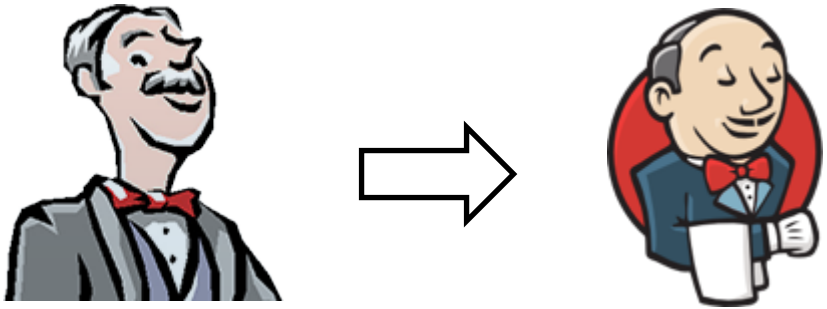
What is Jenkins?

Jenkins is **open source automation server** which can be used to automate all sorts of tasks related to building, testing, and delivering or deploying software.

Jenkins can be installed through native system packages, Docker, or even run standalone by any machine with a Java Runtime Environment (JRE) installed.

History of Jenkins

- Funded by **Kohsuke Kawaguchi** as Hudson CI (in that time he was a developer in Sun Microsystems).
- First Hudson release was in 2004.
- Oracle acquired Sun in 2010 and claimed the right to the ”**Hudson**” name and applied for a trademark in December 2010.
- Kohsuke forked Hudson open source and created the name “**Jenkins”** – the rest is history.

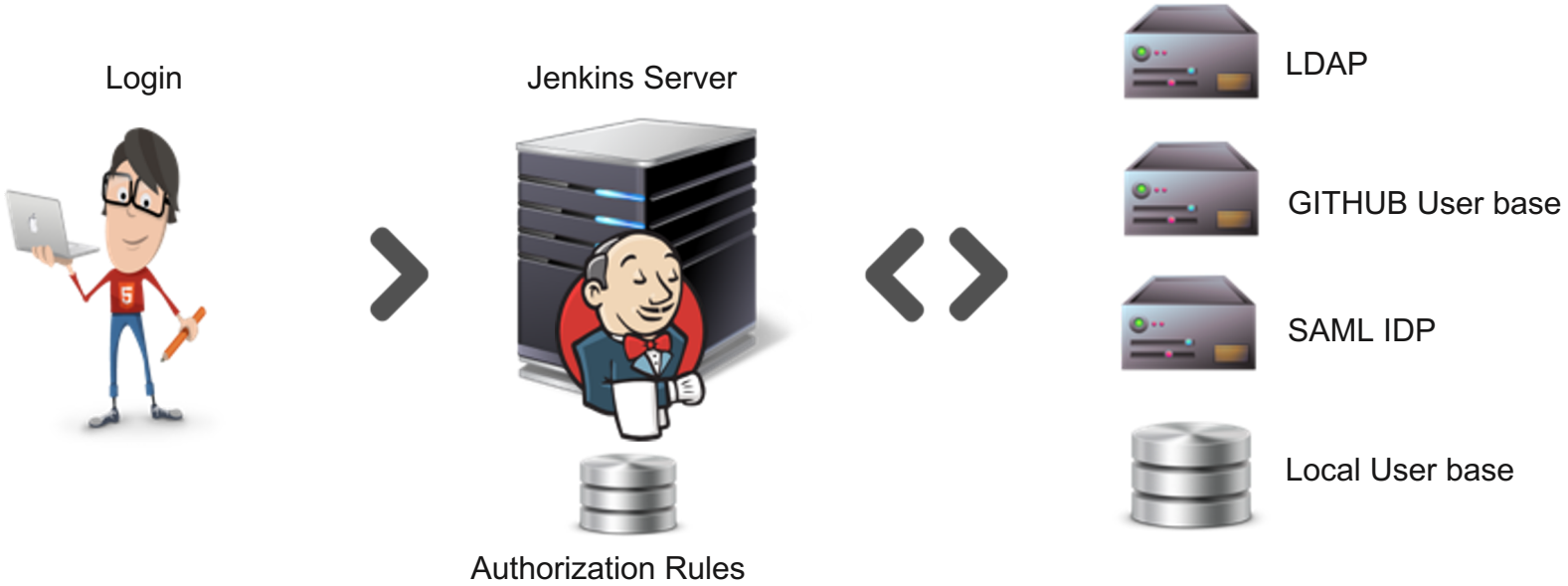


Why Jenkins?

A platform, Free, Open source. Easy to install and easy to use

Jenkins

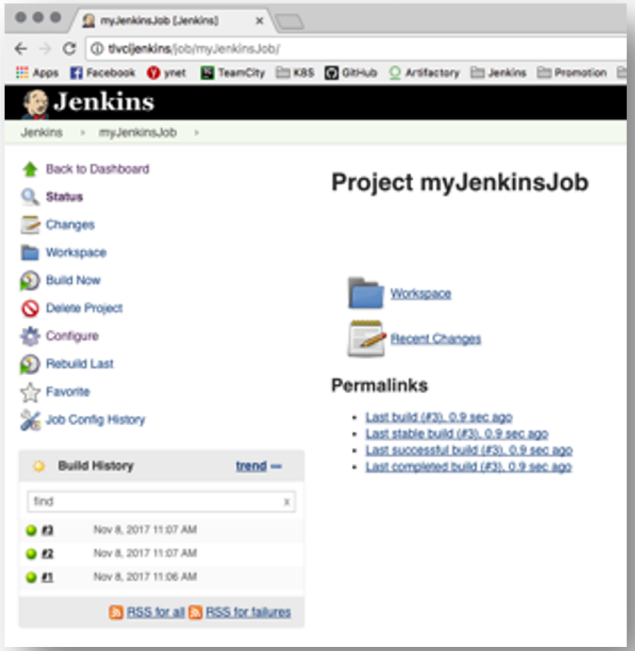
Provides built-in Authentication & Authorization



Jenkins

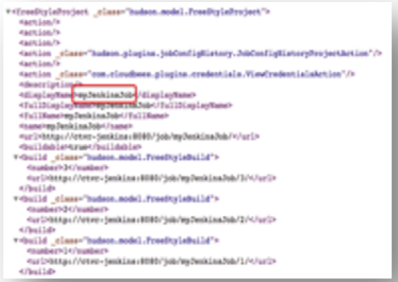
Open platform

1000+ plugins



JSON API

XML API



Jenkins

Extendable

Write your own plugin

Support Groovy DSL

Groovy pipelines

```
Jenkinsfile (Declarative Pipeline)
pipeline {
    agent any

    stages {
        stage('Build') {
            steps {
                sh 'make'
            }
        }
        stage('Test'){
            steps {
                sh 'make check'
                junit 'reports/**/*.xml'
            }
        }
        stage('Deploy') {
            steps {
                sh 'make publish'
            }
        }
    }
}
```

Jenkins

Popular by the community

Jenkins

Short-time (rapid) Development

Practice