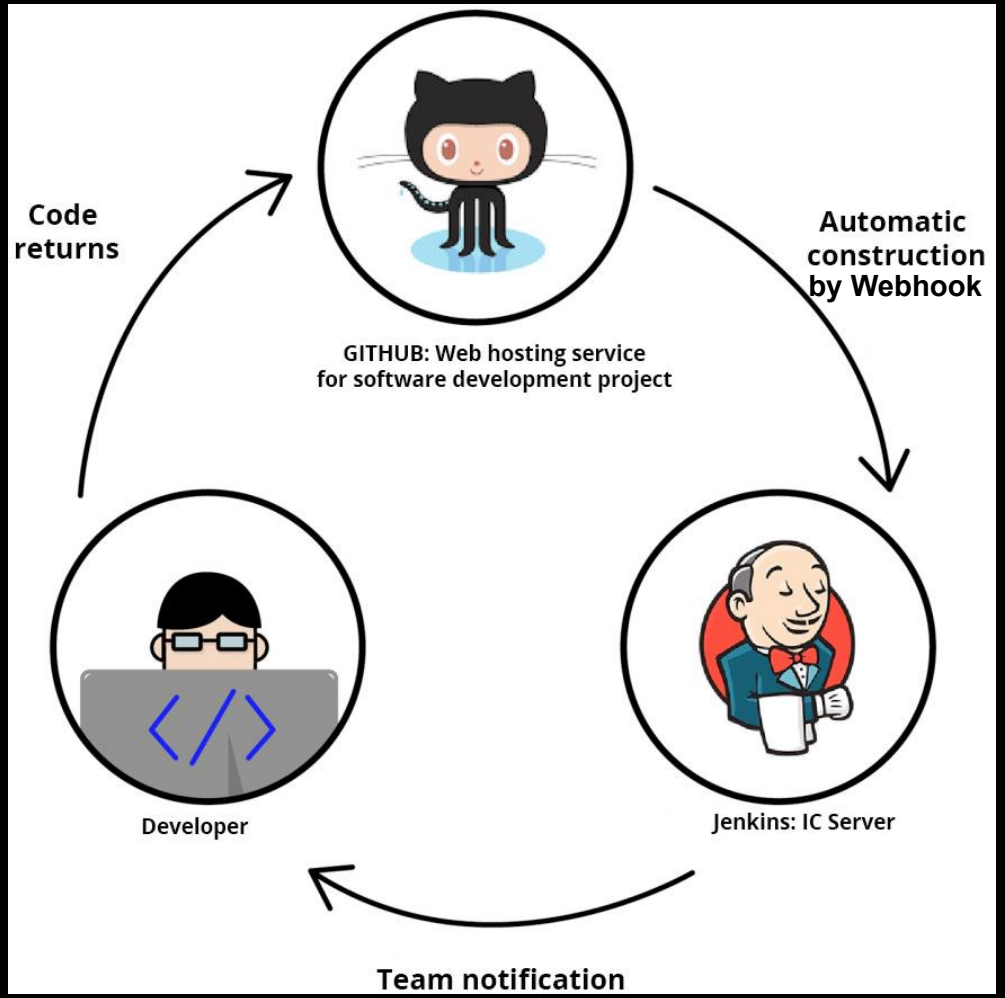
Software Project Management - Laboratory

Lecture n° 19
A.Y. 2020-2021

Prof. Fabrizio Fornari

Continuous Integration with Jenkins

Jenkins triggers a build upon every commit to the source code repository, typically to a development branch.



Build Status

README.md

SPM2020Template

This is a repository for the SPM2020 laboratory course held at the University of Camerino, Computer Science Department.

Especially it provides examples of JUNIT tests, Selenium Tests, and a sort of guide for setting up a github repository.

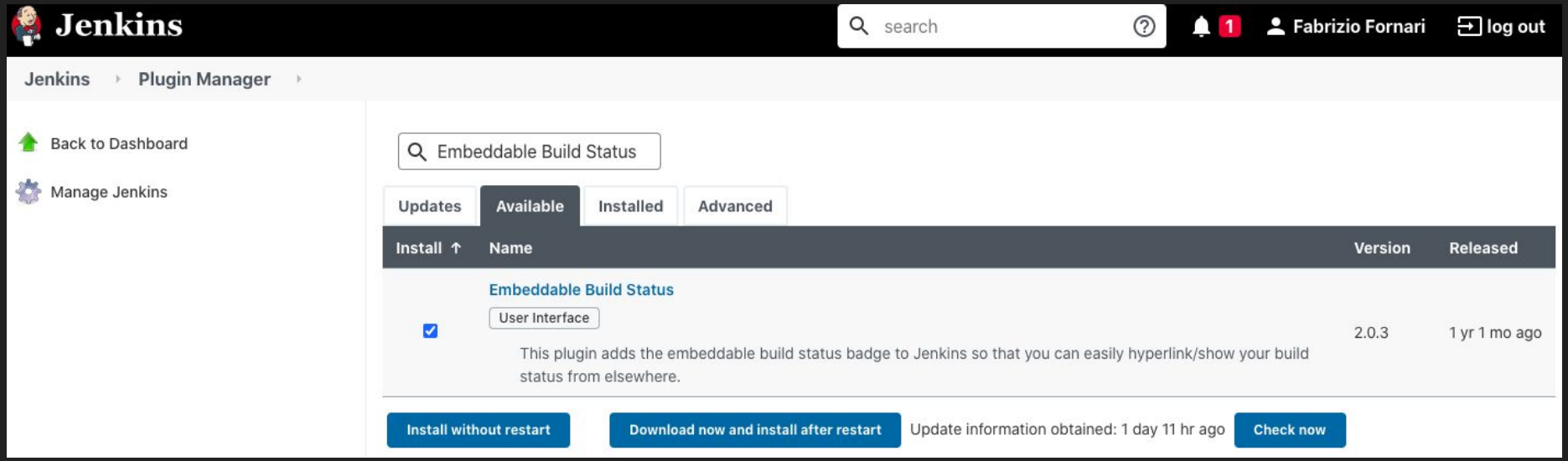
You can git clone it and import it as a Maven project.

https://shields.io

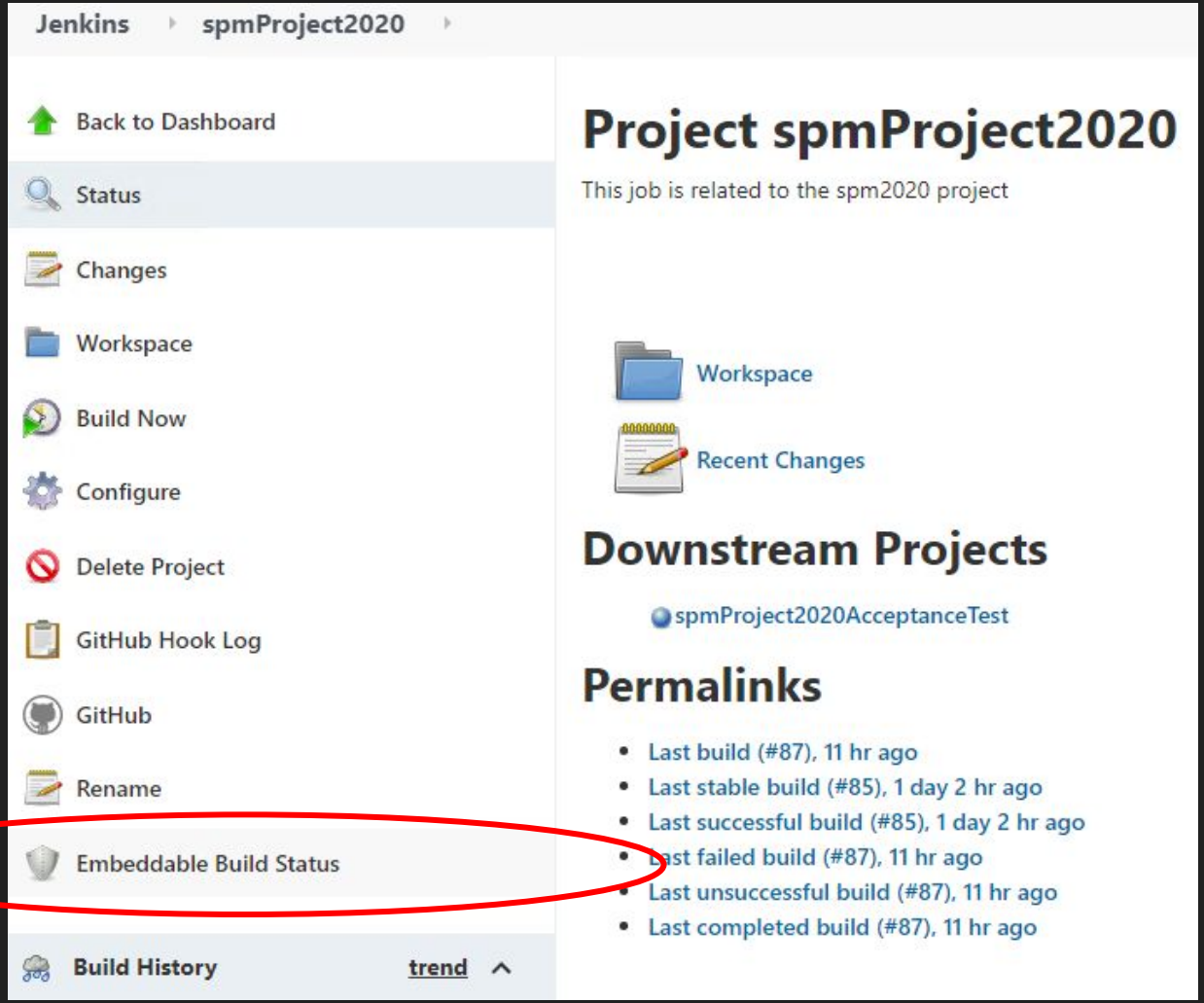
[Jenkins] 

[Travis CI] 

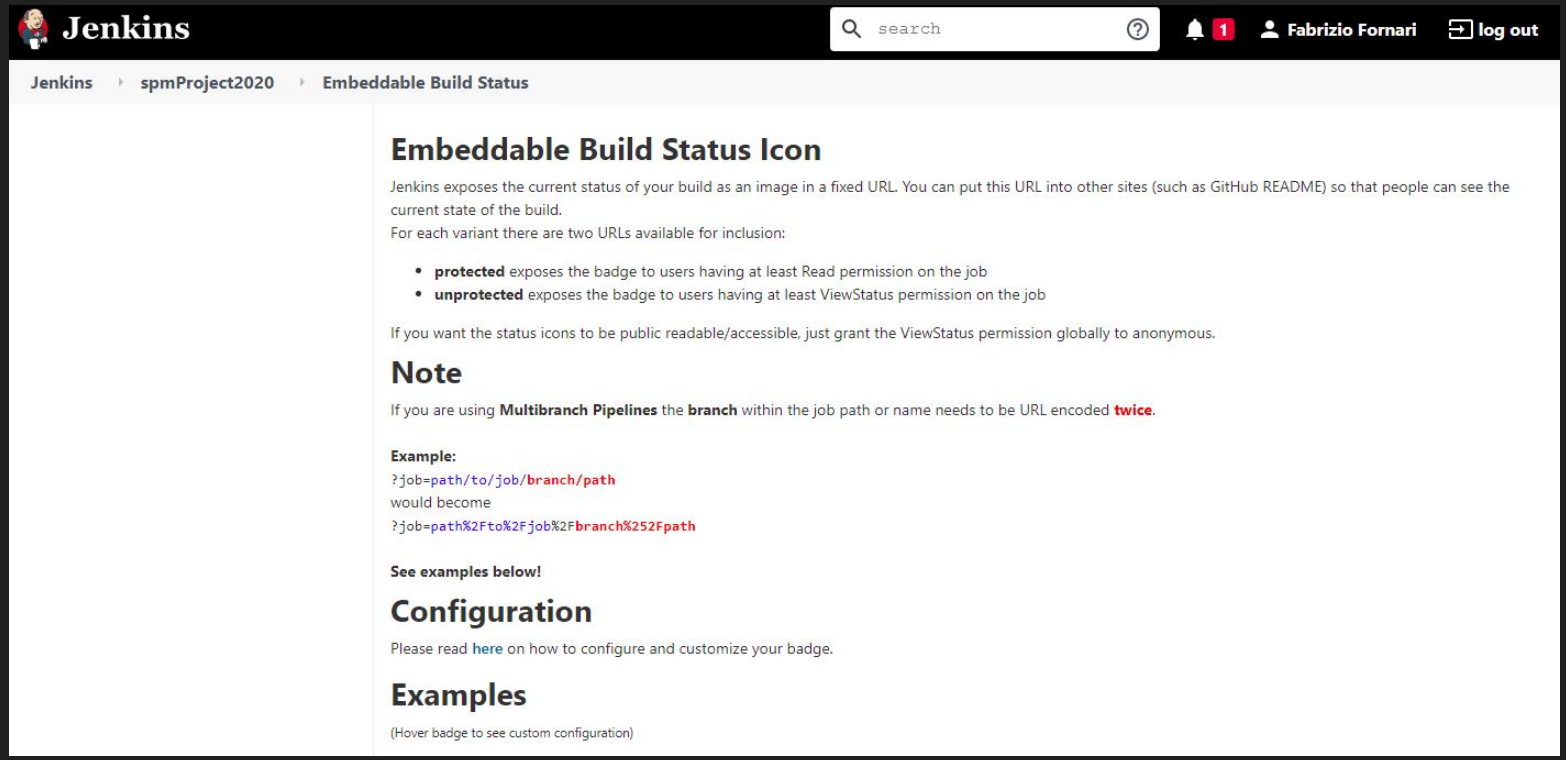
Build Status



Embeddable Build Status

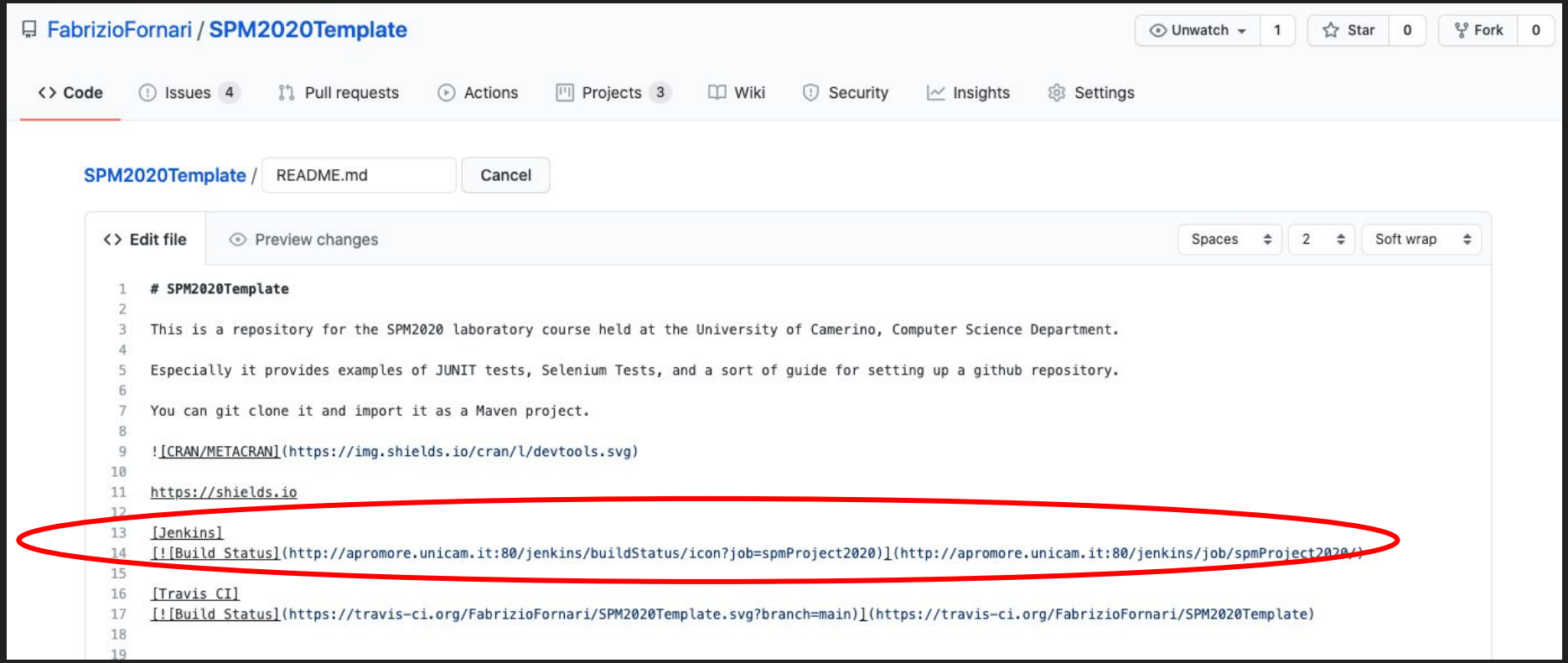


Embeddable Build Status

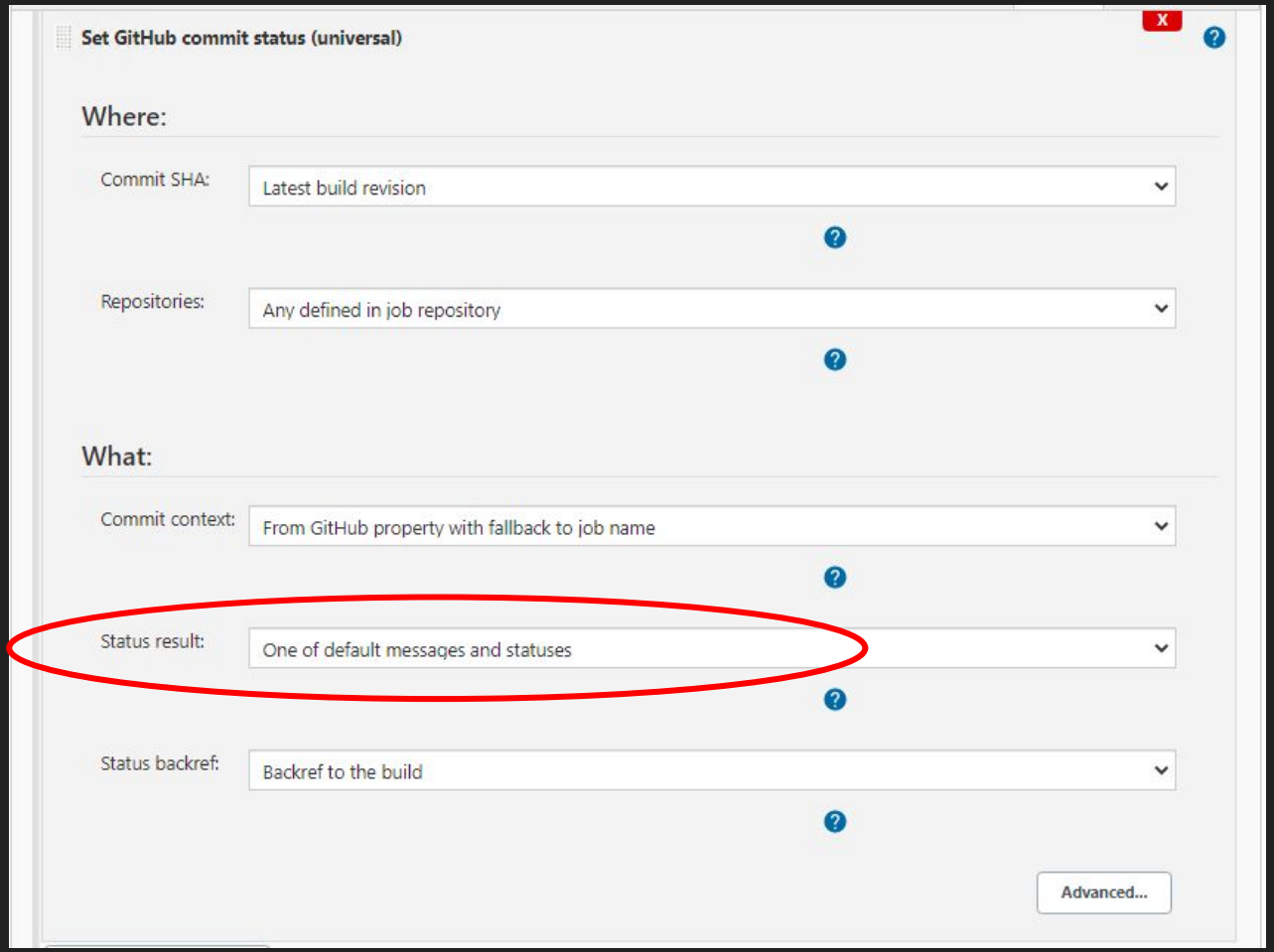


Markdown:
([![Build Status]http://address/jenkins/job/spmProject2020/badge/icon)](http://address/jenkins/job/spmProject2020/)

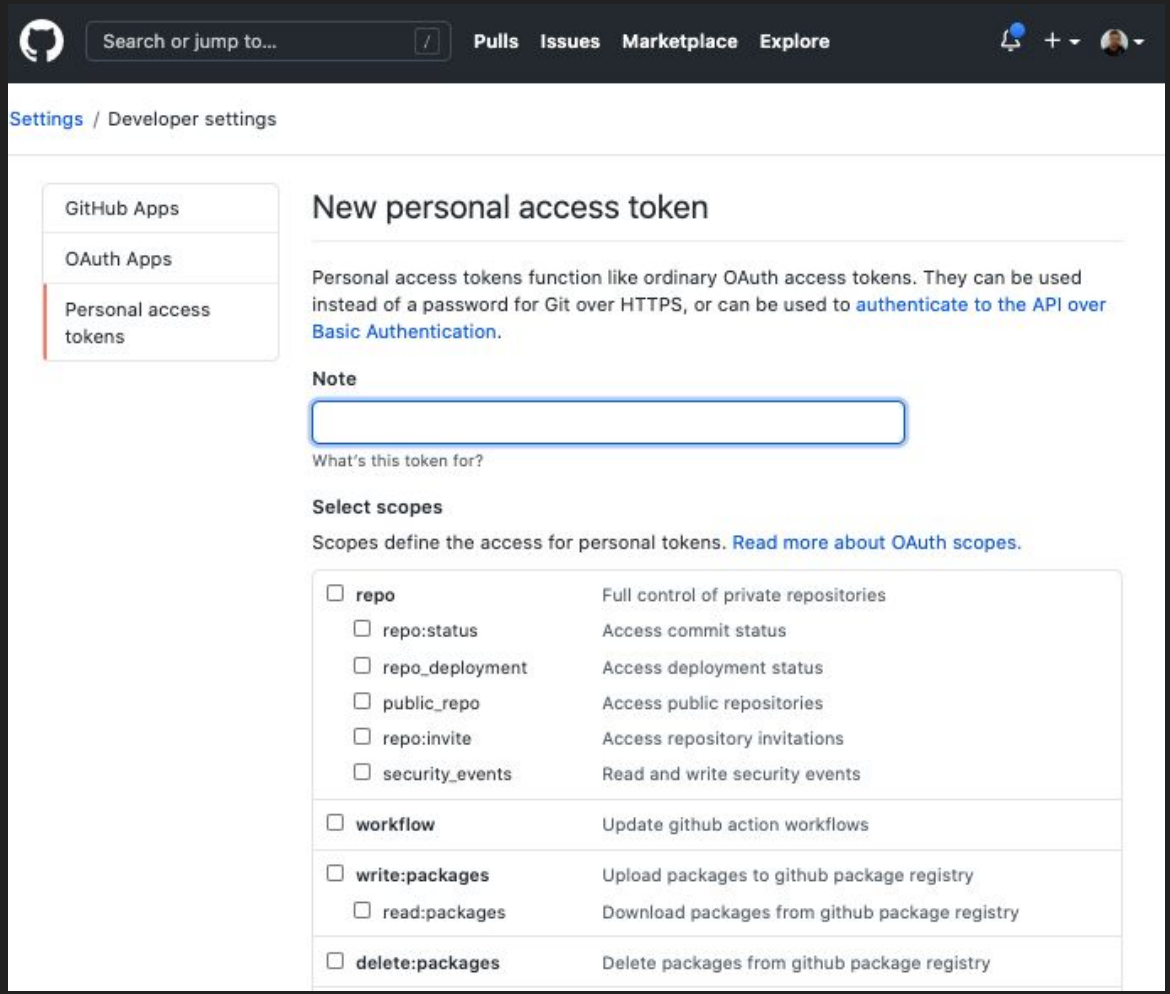
Build Status Markdown on GitHub



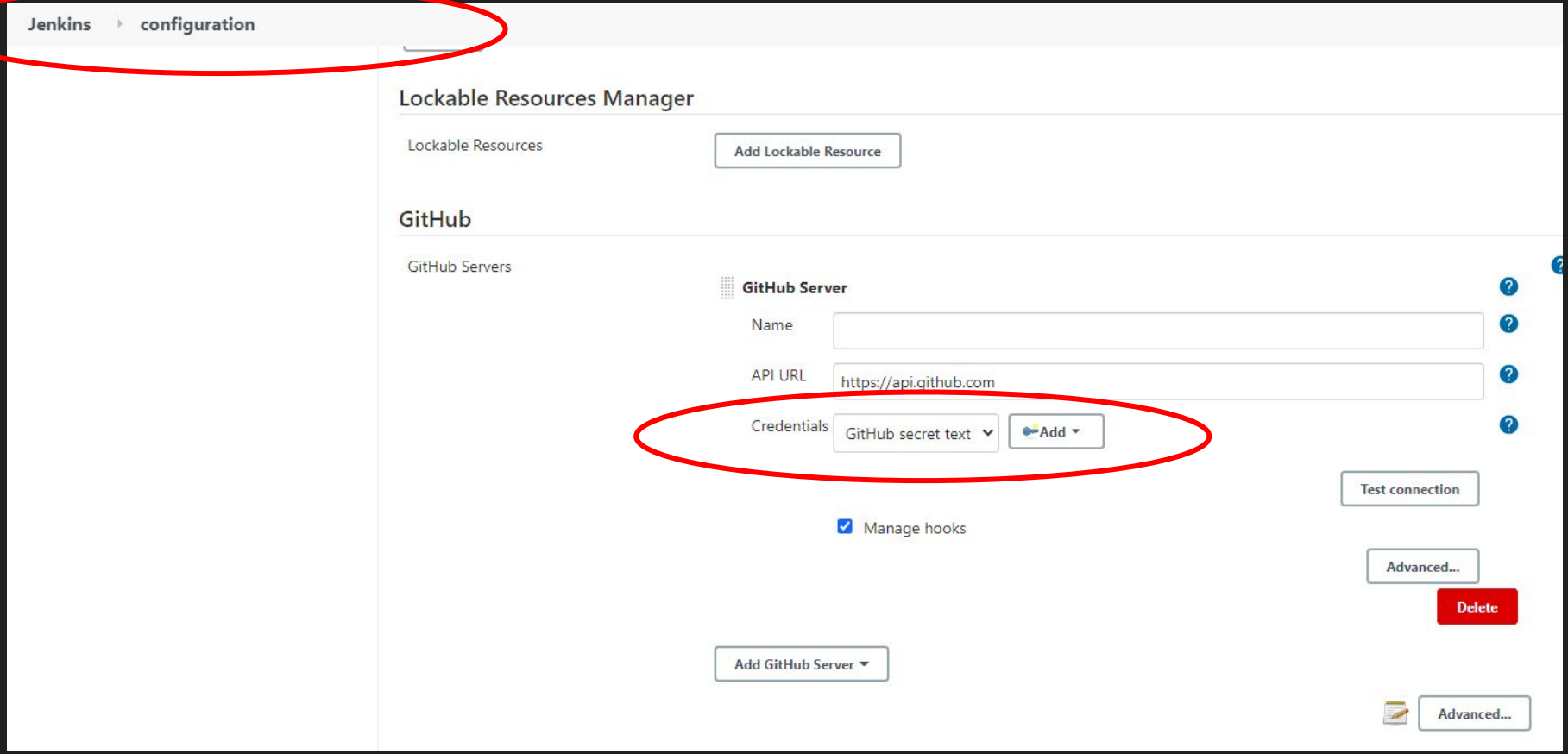
Build Status Configure Job



GitHub Secret Text



GitHub Secret On Jenkins Credentials



Build Status

README.md

SPM2020Template

This is a repository for the SPM2020 laboratory course held at the University of Camerino, Computer Science Department.

Especially it provides examples of JUNIT tests, Selenium Tests, and a sort of guide for setting up a github repository.

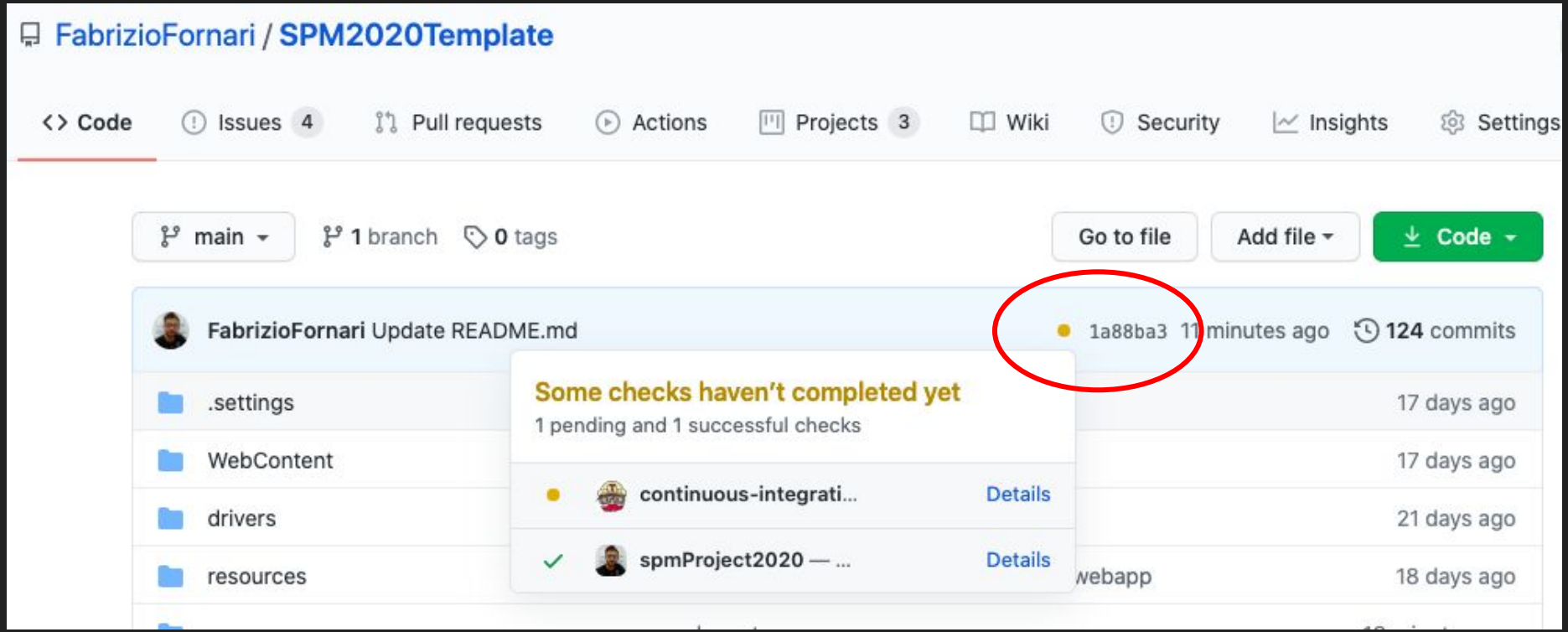
You can git clone it and import it as a Maven project.

https://shields.io

[Jenkins] 

[Travis CI] 

Commit Status

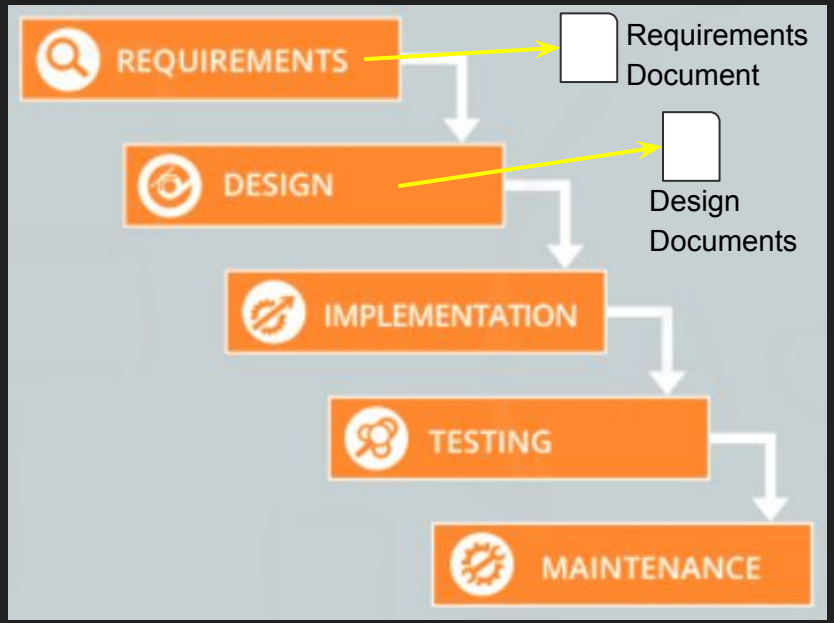


Parenthesis Closed

Software Development Process

Software Development Process is the process of dividing software development work into distinct phases to improve design, product management, and project management. It is also known as a software development life cycle (SDLC)

Waterfall Development Process



The Waterfall Model is document driven.

Each step of the process yields artifacts that are documents.

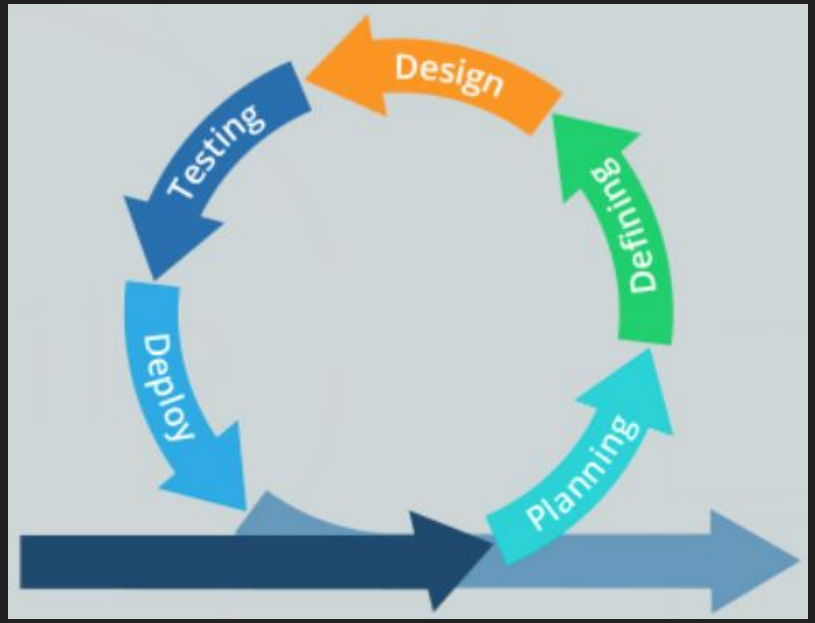
Documents produced during one step are needed for the next step and possibly for later steps.

Managers like the waterfall model because progress is observable and measurable.

Knowledge is lost if team members leave before the project is completed, and it may be difficult for a project to recover from the loss. That is the reason why documents are important.

However, it has repeatedly proven to be very easy to produce impressive documents that eventually prove to be incomplete, inconsistent, hard to consult, or otherwise worthless.

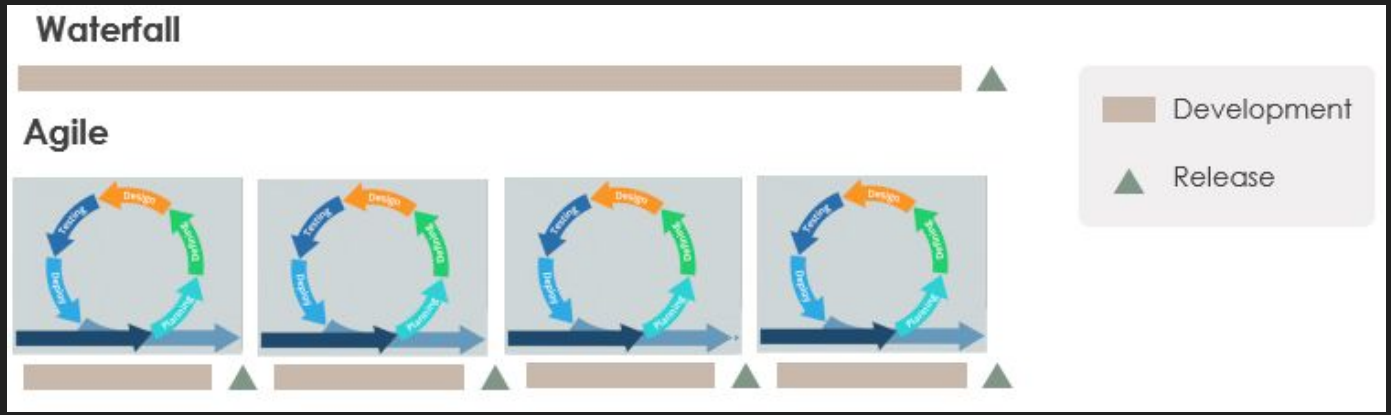
Agile Development Process



Manifesto for Agile Software Development

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

Waterfall vs Agile



Agile characteristics:

- Complex work is divided in simple pieces
- Large organizations are divided into small teams
- Far-reaching project are divided into short time lists of task called sprints

What about your project?

Does it have any kind of documentation?

SCRUM Artifacts

The SCRUM artifacts are used to help define the workload coming into the team and currently being worked upon the team.

The main artifacts:

- Product backlog - a collection of user stories which present functionalities required/wanted by the product team. Usually the product owner takes responsible for this list.

- Sprint backlog - a collection of stories which could be included in the current sprint.

Which kind of documentation in a Agile/Scrum project?

README

You can add a README file to a repository to communicate important information about your project. A README, along with a repository license, contribution guidelines, and a code of conduct, communicates expectations for your project and helps you manage contributions

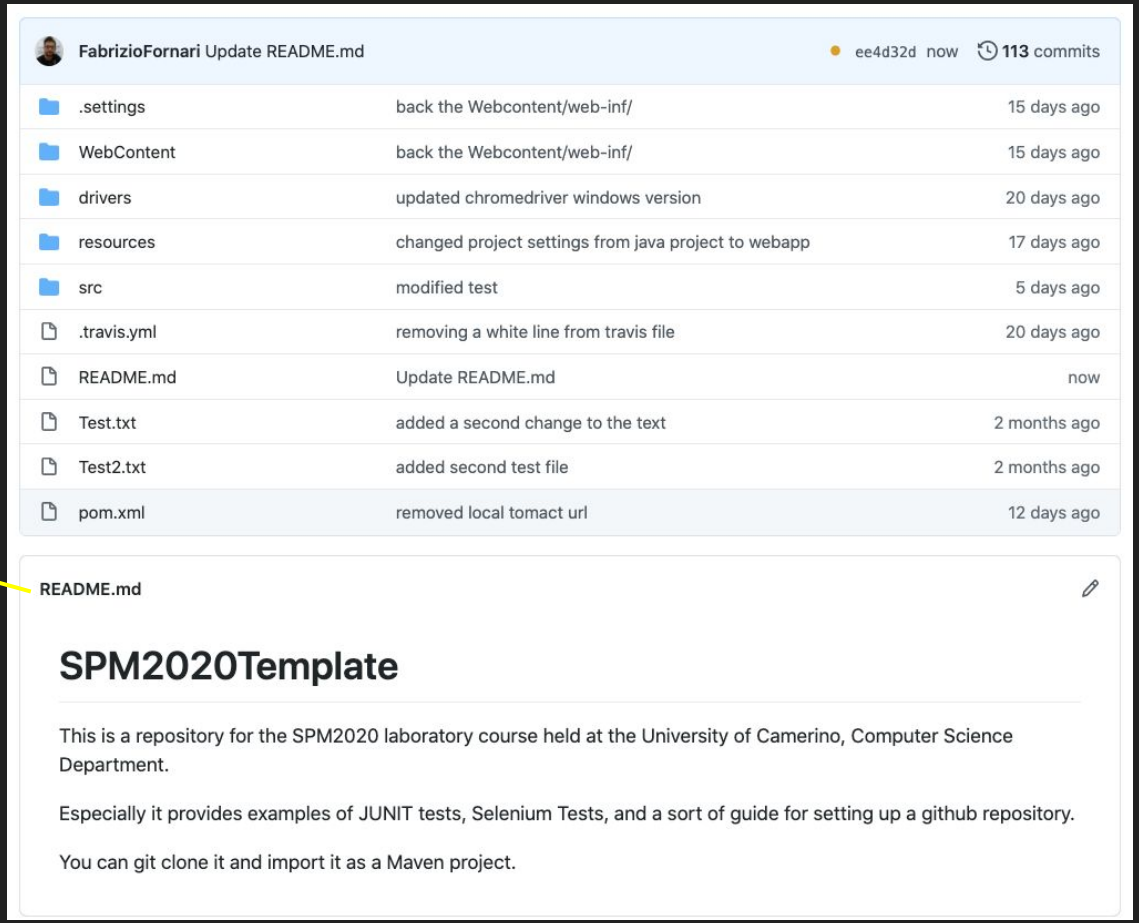
A README is often the first item a visitor will see when visiting your repository. README files typically include information on:

- What the project does
- Why the project is useful
- How users can get started with the project
- Where users can get help with your project
- Who maintains and contributes to the project

If you put your README file in your repository's root, `docs`, or hidden `.github` directory, GitHub will recognize and automatically surface your README to repository visitors.

README

README file



Github - Wiki

Every GitHub repository comes equipped with a section for hosting documentation, called a **wiki**. We can use our repository's wiki to share long-form content about our project, such as how to use it, how we designed it, or its core principles. We can use a wiki to provide additional documentation.

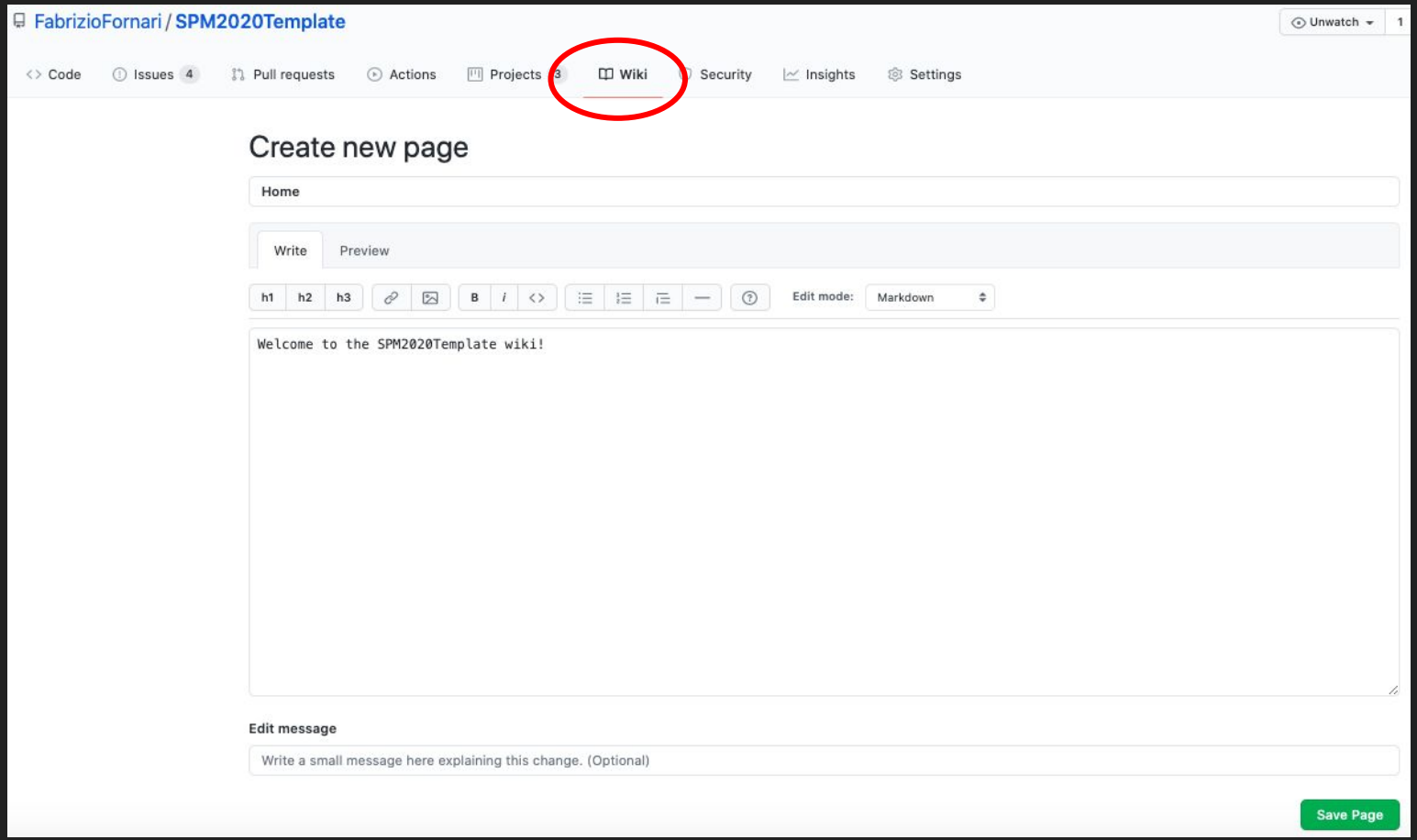
If you create a wiki in a public repository, the wiki is available to the public. If you create a wiki in an internal or private repository, people with access to the repository can also access the wiki.

You can edit wikis directly on GitHub, or you can edit wiki files locally. By default, only people with write access to your repository can make changes to wikis, although you can allow everyone on GitHub to contribute to a wiki in a public repository.

Cloning wikis to your computer

```
$ git clone https://github.com/YOUR_USERNAME/YOUR_REPOSITORY.wiki.git
# Clones the wiki locally
```

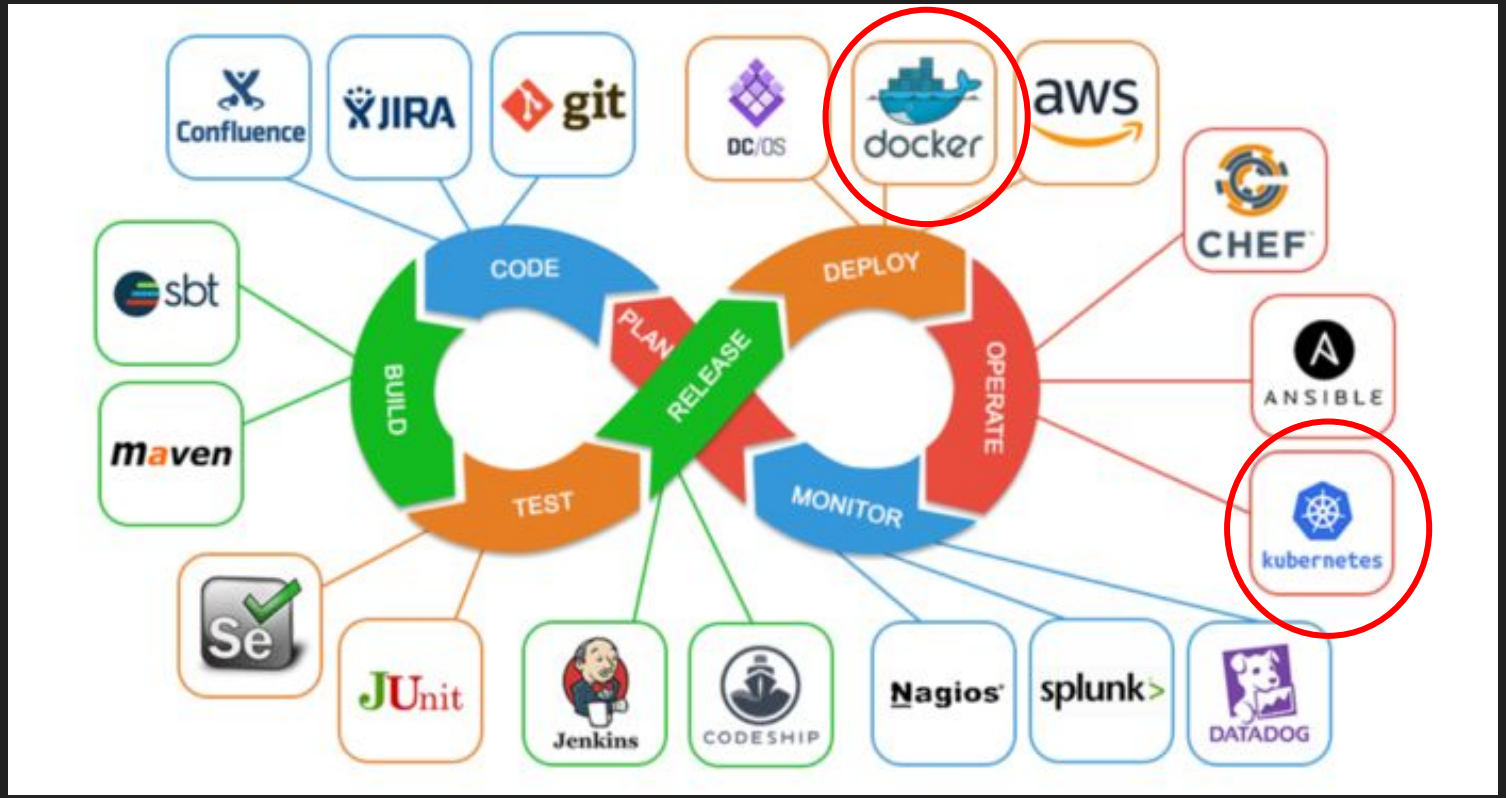
Github - Wiki

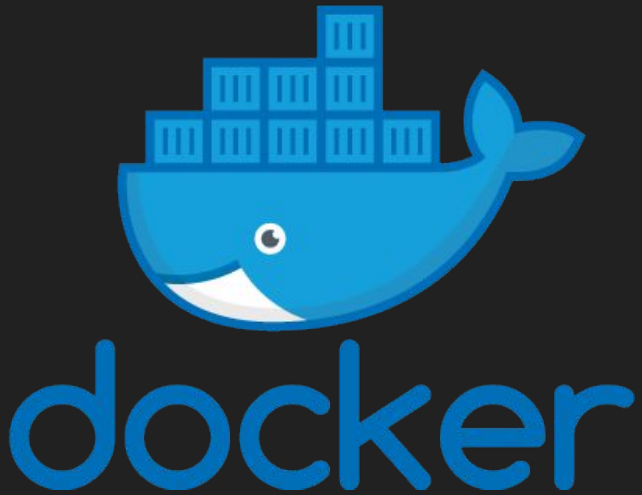

FabrizioFornari / SPM2020Template
Unwatch
1
Code
Issues 4
Pull requests
Actions
Projects 3
Wiki
Security
Insights
Settings
Create new page
Home
Write
Preview
h1
h2
h3
B
i
Edit mode:
Markdown
Welcome to the SPM2020Template wiki!
Edit message
Write a small message here explaining this change. (Optional)
Save Page

Github - Wiki Permissions

Looking ahead...

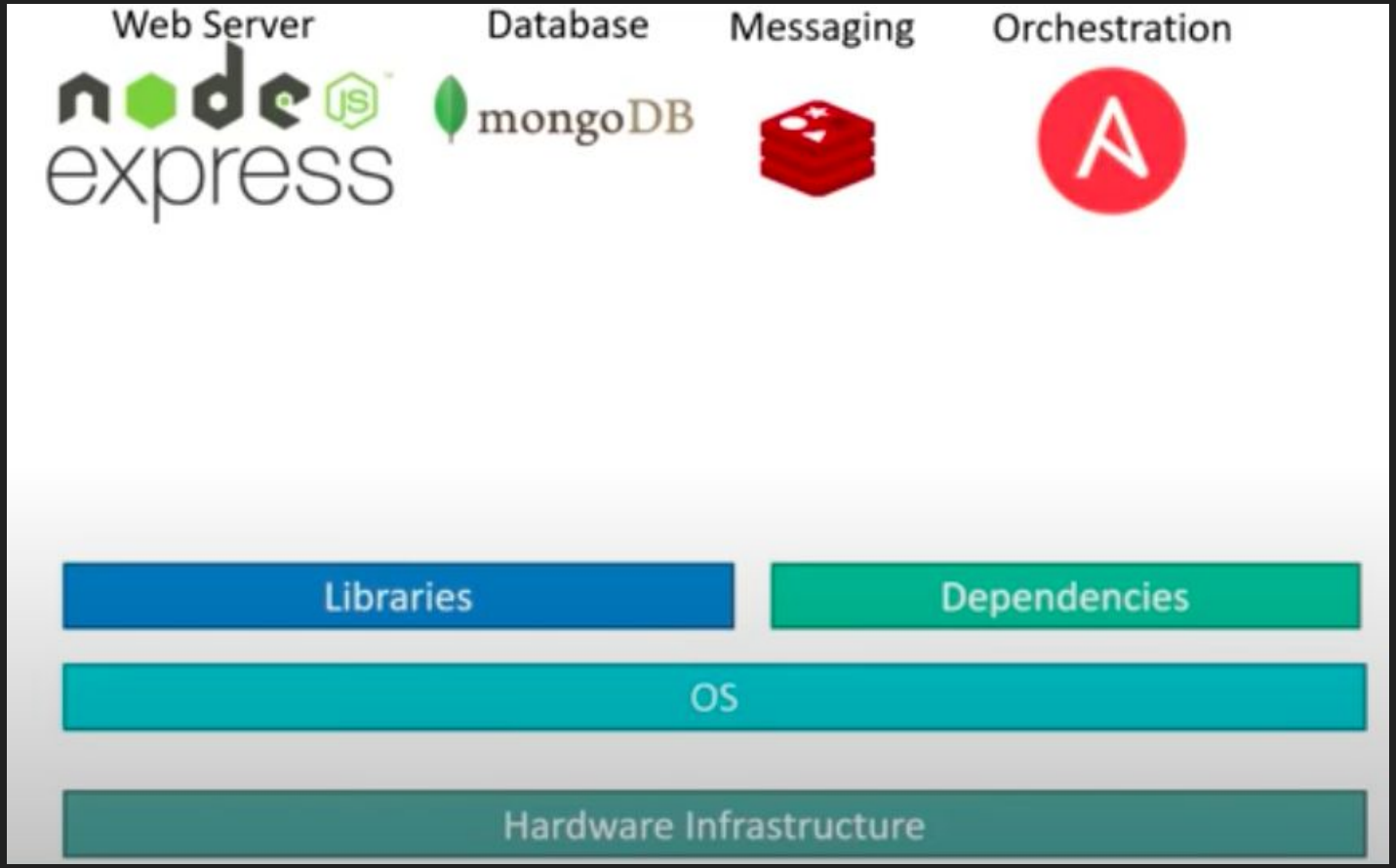
DevOps Technologies





https://www.docker.com/

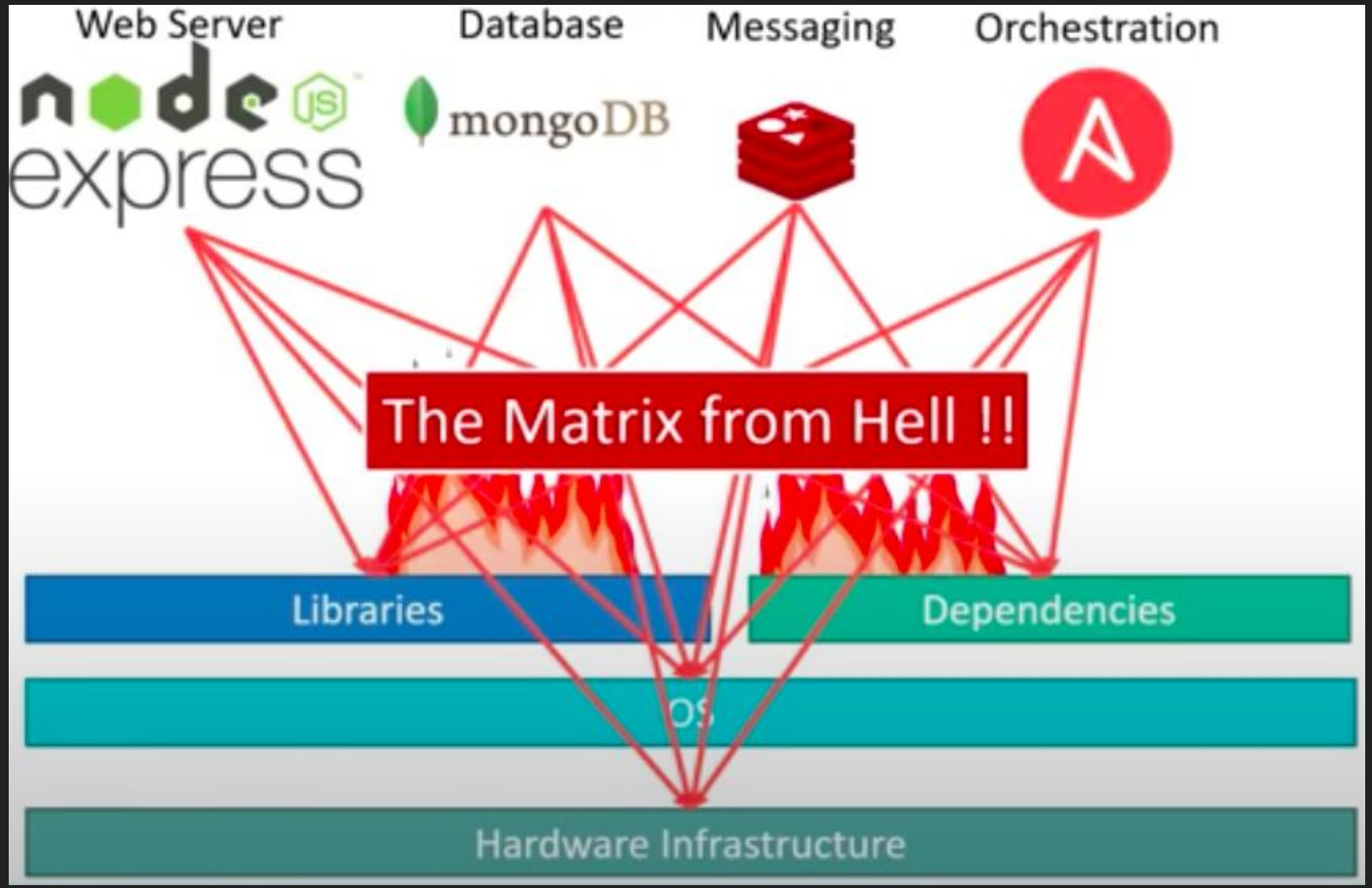
Docker is an open platform for developing, shipping, and running applications.
Docker enables you to separate your applications from your infrastructure so you can deliver software quickly.
With Docker, you can manage your infrastructure in the same ways you manage your applications.


Web Server
Database
Messaging
Orchestration
node
express
mongoDB
Libraries
Dependencies
OS
Hardware Infrastructure

Compatibility/ Dependency

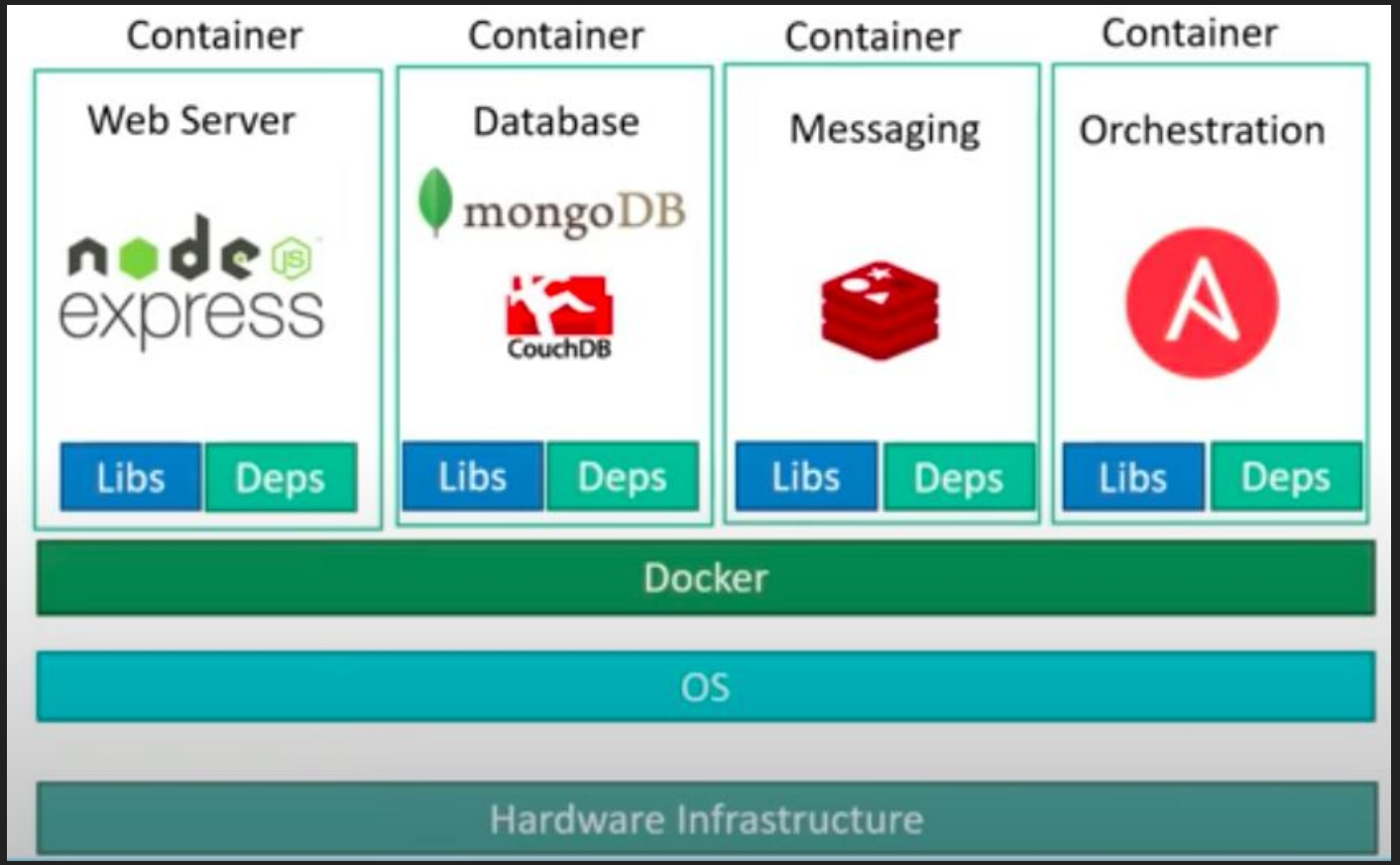
Long setup time

Different Dev/Test/Prod environments



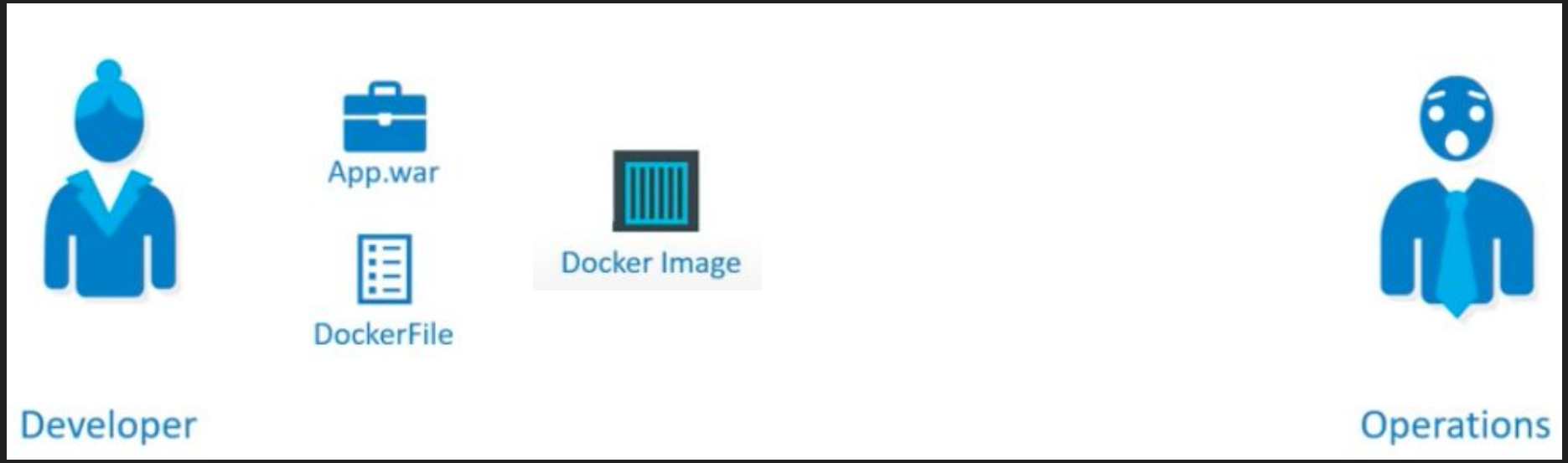
Containerized Application

Run each service with its own dependencies in separate containers



From Application to Container

From Application to Container



It Fixes the traditional “but it works on my machine”

From Application to Container

It Fixes the traditional “but it works on my machine”

Public Docker Images Repository

https://hub.docker.com/

Jenkins + Docker

Jenkins

docker

...so a Docker Host

What if a Docker Host fails?

Orchestrating Hosts

Orchestration technology focuses on clustering and managing containers and hosts.

Docker Swarm: Easy to setup but lacks autoscaling

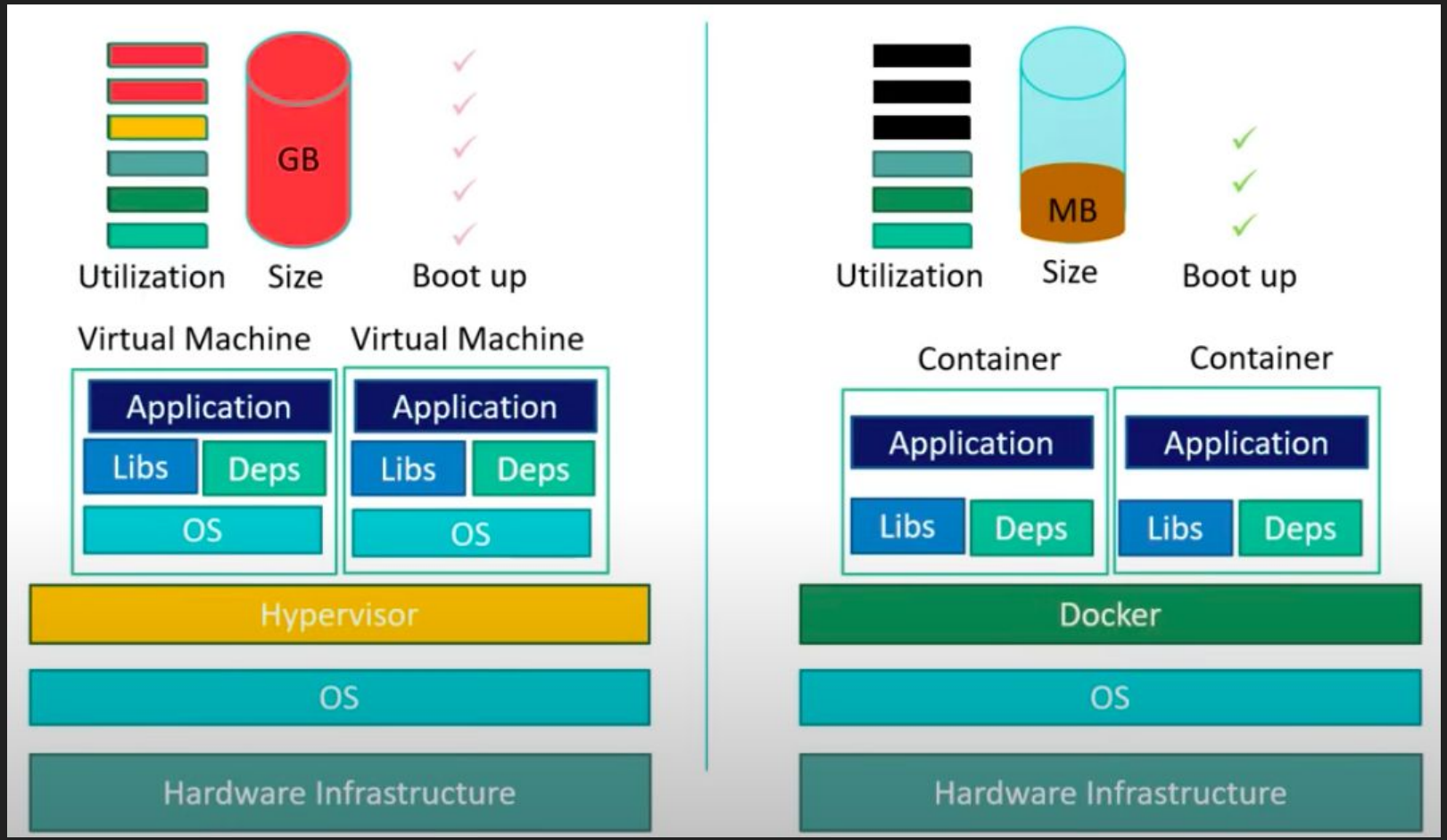
Kubernetes: from Google, difficult to setup but supports many advanced features, all public cloud supports it

MESOS: from Apache, difficult to setup but supports many advanced features,

Kubernetes

A fundamental difference between Kubernetes and Docker is that Kubernetes is meant to run across a cluster while Docker runs on a single node. Kubernetes is more extensive than Docker Swarm and is meant to coordinate clusters of nodes at scale in production in an efficient manner.

VMs vs Containers



What’s next?

Date	Topic
18/12/2020	Project Status Check with Groups
07/01/2020	Review of the Entire Course
08/01/2020	Sprint Review / Final Project Discussion

Course Overview

Course Objective

The course introduced the students to the basic knowledge of complex software system production following the **DevOps methodology**.

Learning Outcome

The student will be able to manage the organization and the development of a software applying DevOps methodology.

Overview

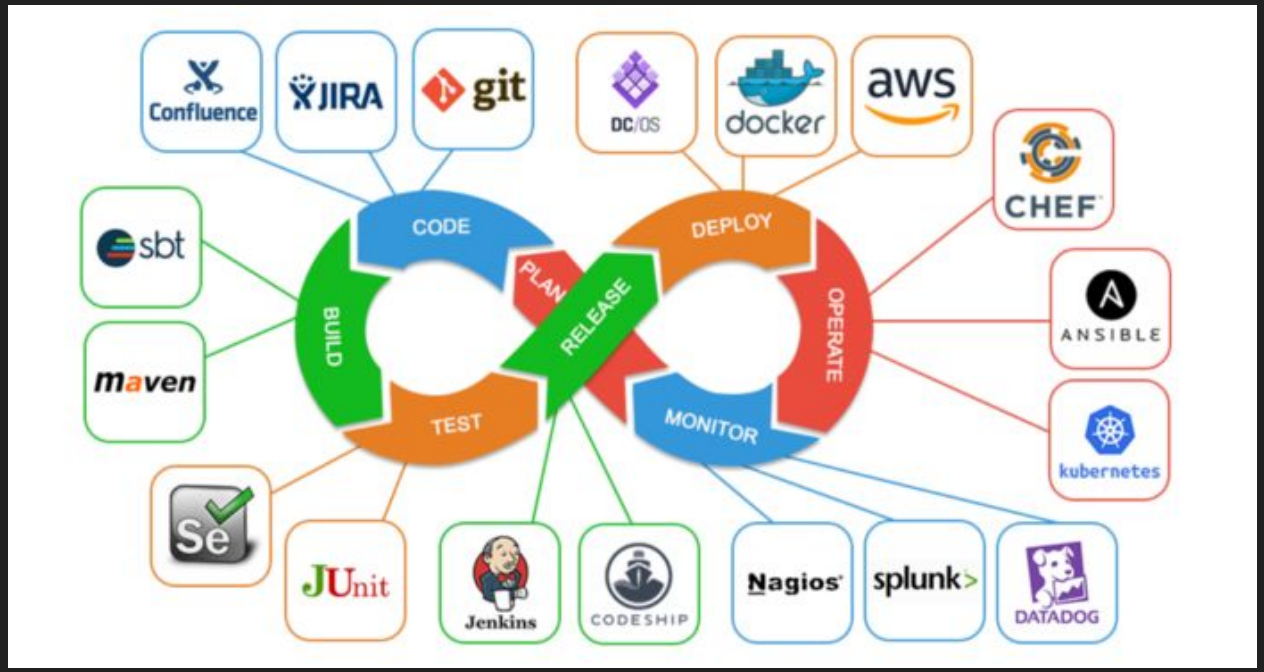
Website:

- http://didattica.cs.unicam.it/doku.php?id=didattica:magistrale:spm:ay_2021:main&s[]=spm

You can contact me for any question related to the course: *fabrizio.fornari@unicam.it*

Note: only email coming from the @studenti.unicam.it domain will be processed.

What's next?


Confluence
JIRA
git
DC/OS
docker
aws
CHEF
sbt
maven
Se
JUnit
Jenkins
CODESHIP
Nagios
splunk>
DATADOG
kubernetes
ANSIBLE
CODE
PLAN
BUILD
TEST
RELEASE
DEPLOY
OPERATE
MONITOR


PROS Lab
pros.unicam.it

My Topics

Business Process Management
Business Process Modeling and Verification
Business Process and IoT
Process Mining
Software Project Management

Tools:

- BProVe, Business Process Verifier
- BEBoP, understandaBility vErifier for Business Process models
- RePROSitory, Repository of open PROcess modelS

Group Projects or Thesis

I supervise group projects and experimental thesis.

I try to apply together with the students the methodology and tools that we have seen during the course.

You can contact me for any question related to the course and for additional information about projects and thesis: *fabrizio.fornari@unicam.it*

Note: only email coming from the @studenti.unicam.it domain will be processed.