Software Project Management - Laboratory

Lecture n° 17
A.Y. 2020-2021

Prof. Fabrizio Fornari

Jenkins

Jenkins is used to build and test your product continuously, so developers can continuously integrate changes into the build.

https://jenkins.io/

Apache Tomcat

The Apache Tomcat® software is an open source implementation of the Java Servlet, JavaServer Pages, Java Expression Language and Java WebSocket technologies.

http://tomcat.apache.org/

Download Tomcat

https://tomcat.apache.org/download-90.cgi

Jenkins on Tomcat

Download the latest .war file

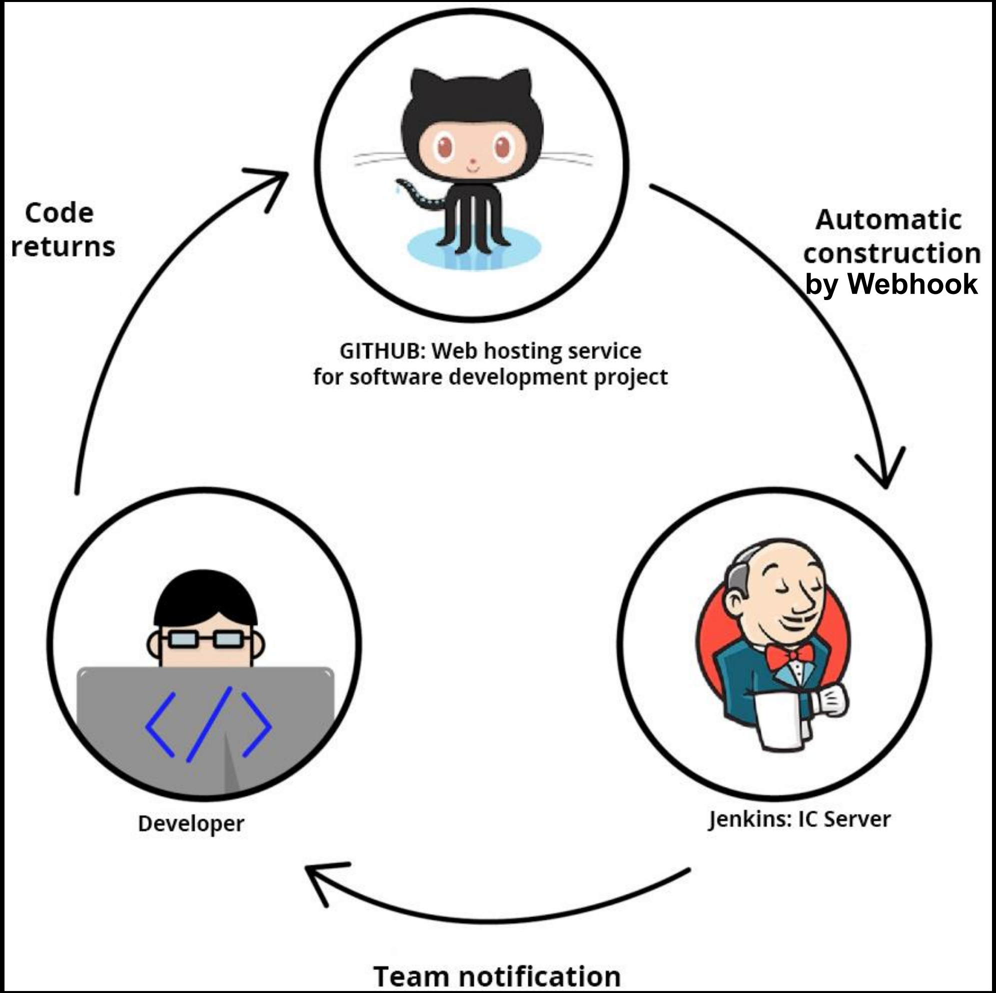
https://www.jenkins.io/download/

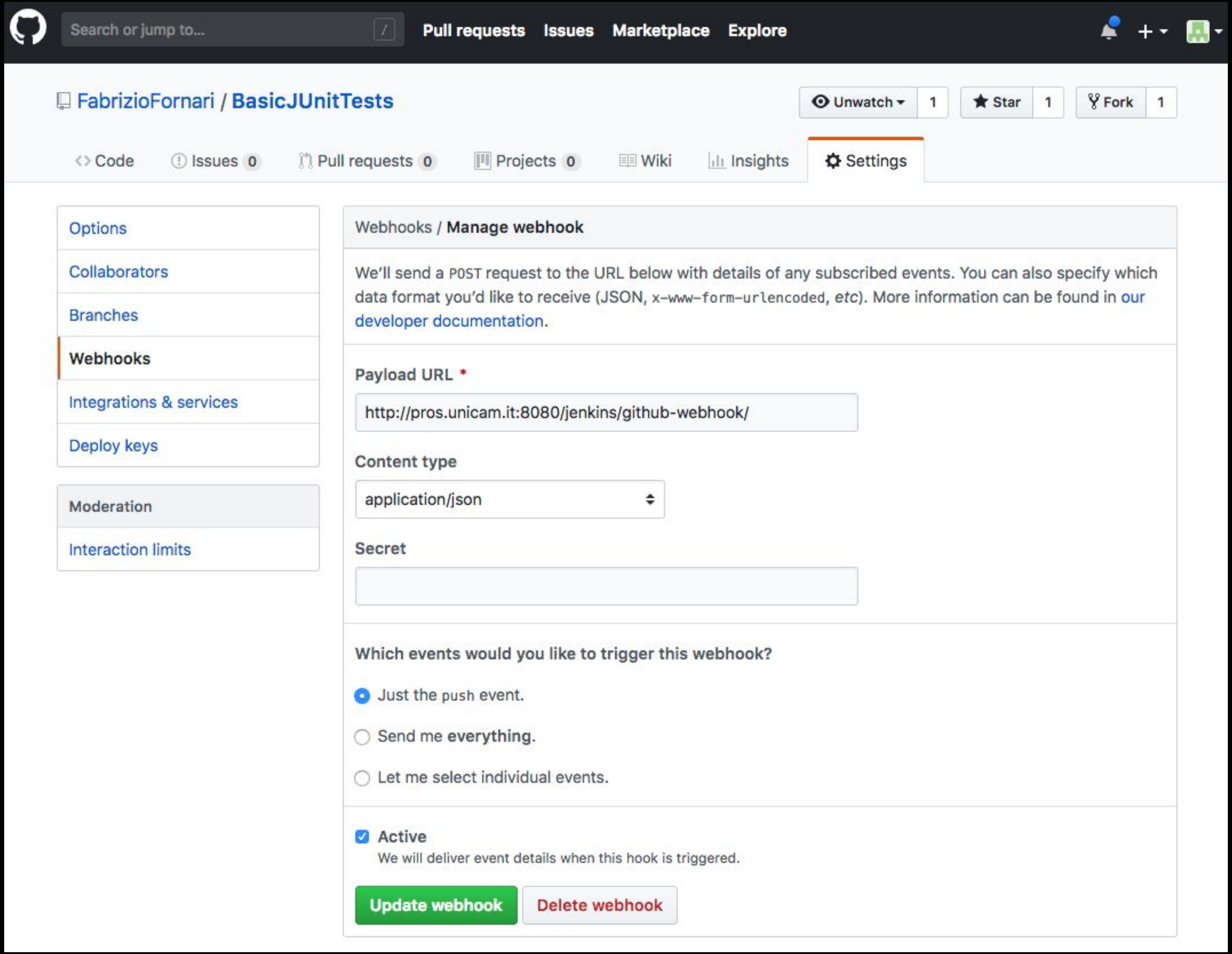
Deploy the .war on Tomcat

https://www.jenkins.io/doc/book/installing/#war-file

Continuous Integration with Jenkins

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


Code returns
GITHUB: Web hosting service for software development project
Automatic construction by Webhook
Developer
Jenkins: IC Server
Team notification



Webhooks / **Manage webhook**

We'll send a `POST` request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, `x-www-form-urlencoded`, *etc*). More information can be found in our developer documentation.

Payload URL *

http://pros.unicam.it:8080/jenkins/github-webhook/

Content type

application/json

Secret

Which events would you like to trigger this webhook?

● Just the `push` event.

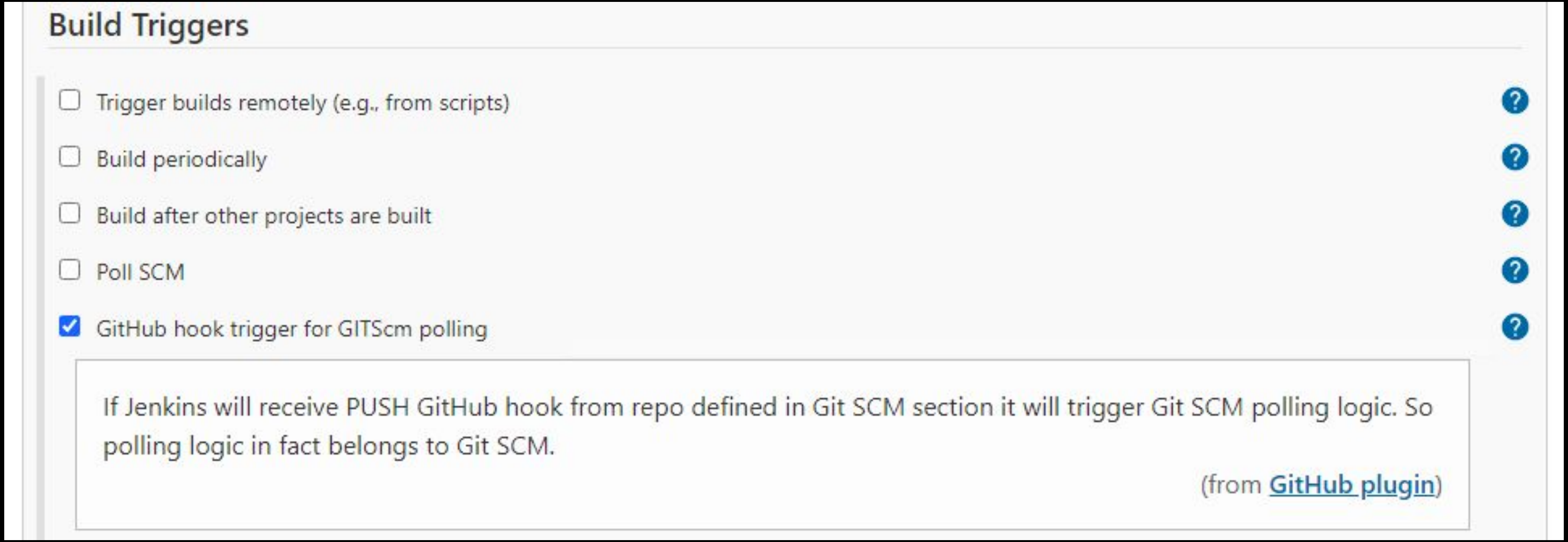
○ Send me **everything**.

○ Let me select individual events.

☑ **Active**

We will deliver event details when this hook is triggered.

Jenkins GitHub hook trigger



Build Triggers

- ☐ Trigger builds remotely (e.g., from scripts)
- ☐ Build periodically
- ☐ Build after other projects are built
- ☐ Poll SCM
- ☑ GitHub hook trigger for GITScm polling

If Jenkins will receive PUSH GitHub hook from repo defined in Git SCM section it will trigger Git SCM polling logic. So polling logic in fact belongs to Git SCM.

(from GitHub plugin)

Automatically Deploy a Web App

Automatically deploy a WAR on Tomcat with Jenkins

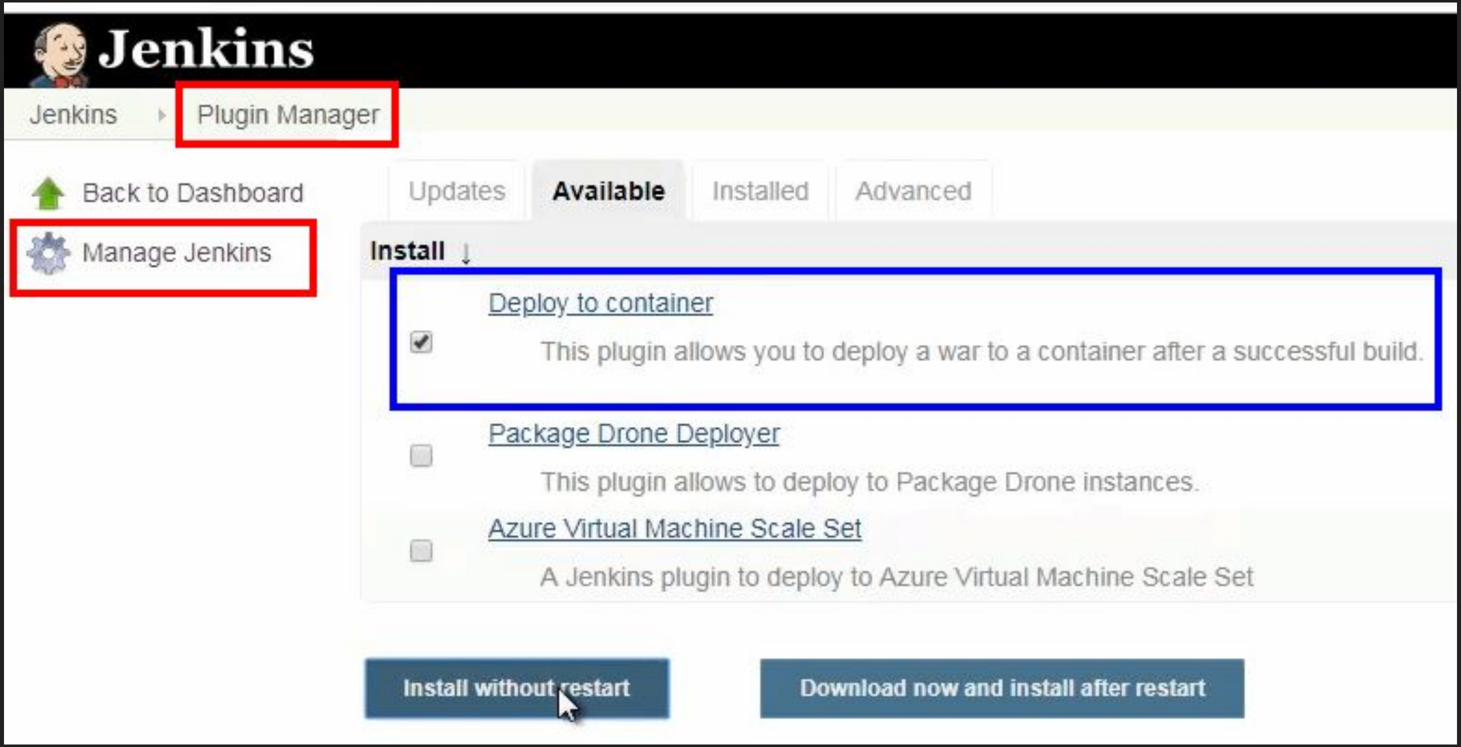
To perform a Maven Tomcat deploy of a WAR file you must first set up a user in Tomcat with the appropriate rights. You can do this with an edit of the tomcat-users.xml file, which can be found in Tomcat's conf sub-directory. Add the following entry inside the tomcat-users tag:

```
<!--  User to deploy WAR file to Tomcat from Maven -->
<user username="war-deployer" password="jenkins-tomcat-plugin"
            roles="manager-gui, manager-script, manager-jmx" />
```

Save the tomcat-users.xml file and restart the server to have the changes take effect.

Automatically deploy a WAR on Tomcat with Jenkins

Install the Deploy to container plugin

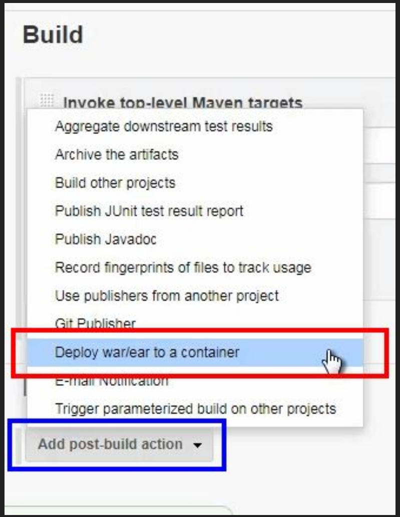


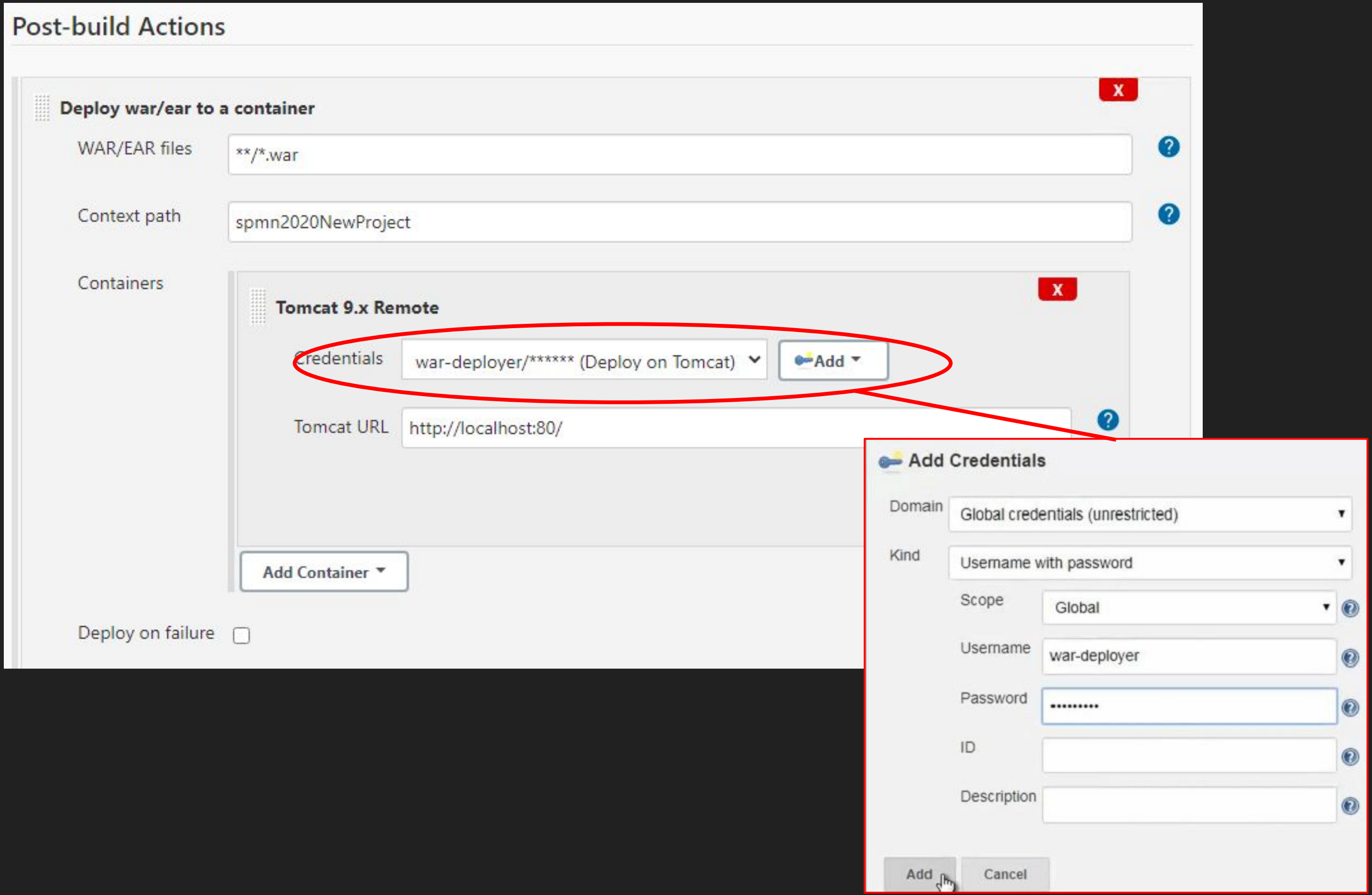
☑ Deploy to container
This plugin allows you to deploy a war to a container after a successful build.

□ Package Drone Deployer
This plugin allows to deploy to Package Drone instances.

□ Azure Virtual Machine Scale Set
A Jenkins plugin to deploy to Azure Virtual Machine Scale Set

Automatically deploy a WAR on Tomcat with Jenkins


Build
Invoke top-level Maven targets
Aggregate downstream test results
Archive the artifacts
Build other projects
Publish JUnit test result report
Publish Javadoc
Record fingerprints of files to track usage
Use publishers from another project
Git Publisher
Deploy war/ear to a container
E-mail Notification
Trigger parameterized build on other projects
Add post-build action



Post-build Actions

Deploy war/ear to a container

WAR/EAR files: `**/*.war`

Context path: spmn2020NewProject

Containers

Tomcat 9.x Remote

Credentials: war-deployer/****** (Deploy on Tomcat)

Tomcat URL: http://localhost:80/

Deploy on failure □

Add Credentials

Domain: Global credentials (unrestricted)

Kind: Username with password

Scope: Global

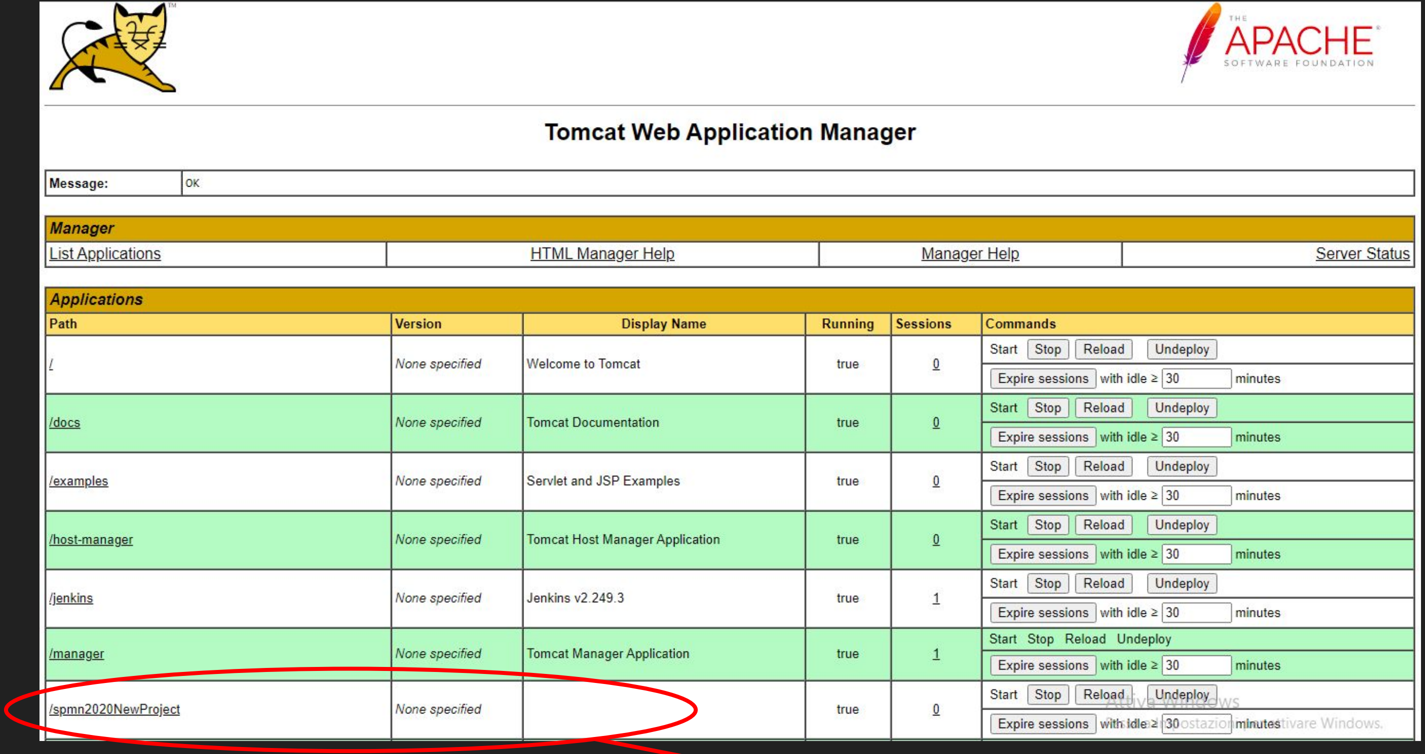
Username: war-deployer

Password: ••••••••••

ID:

Description:

Automatically deploy a WAR on Tomcat with Jenkins



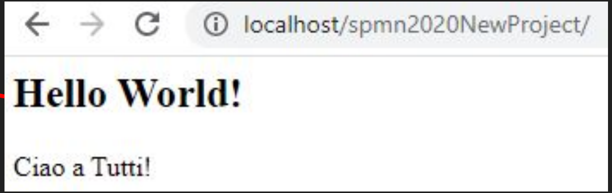
Tomcat Web Application Manager

Message:	OK

Manager			
List Applications	HTML Manager Help	Manager Help	Server Status

Applications

Path	Version	Display Name	Running	Sessions	Commands
/	*None specified*	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/docs	*None specified*	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/examples	*None specified*	Servlet and JSP Examples	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/host-manager	*None specified*	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/jenkins	*None specified*	Jenkins v2.249.3	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/manager	*None specified*	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/spmn2020NewProject	*None specified*		true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes



localhost/spmn2020NewProject/

Hello World!

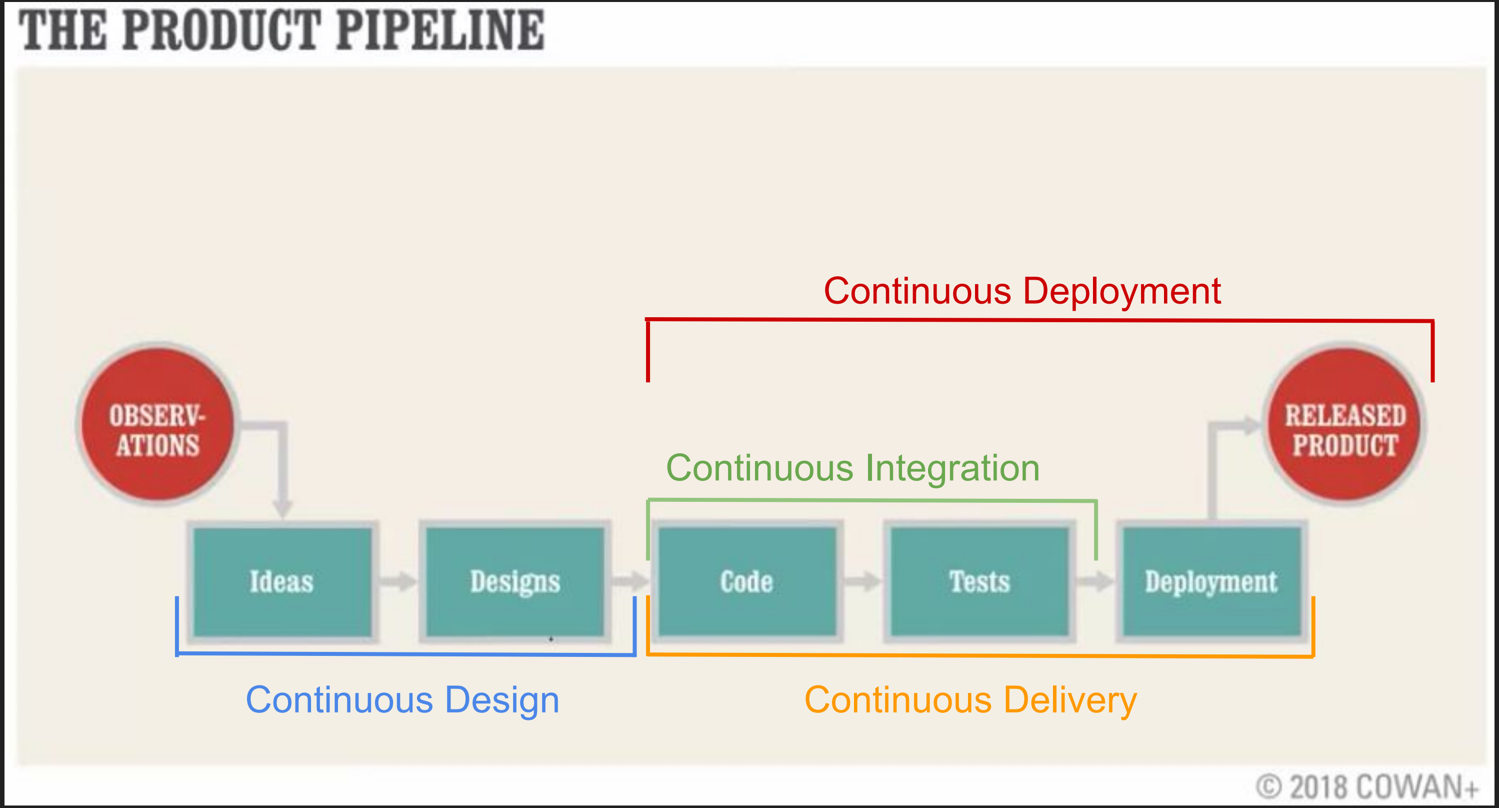
Ciao a Tutti!

Running Selenium Tests

Run Test with Chrome
(MyWebappSeleniumTest.java)

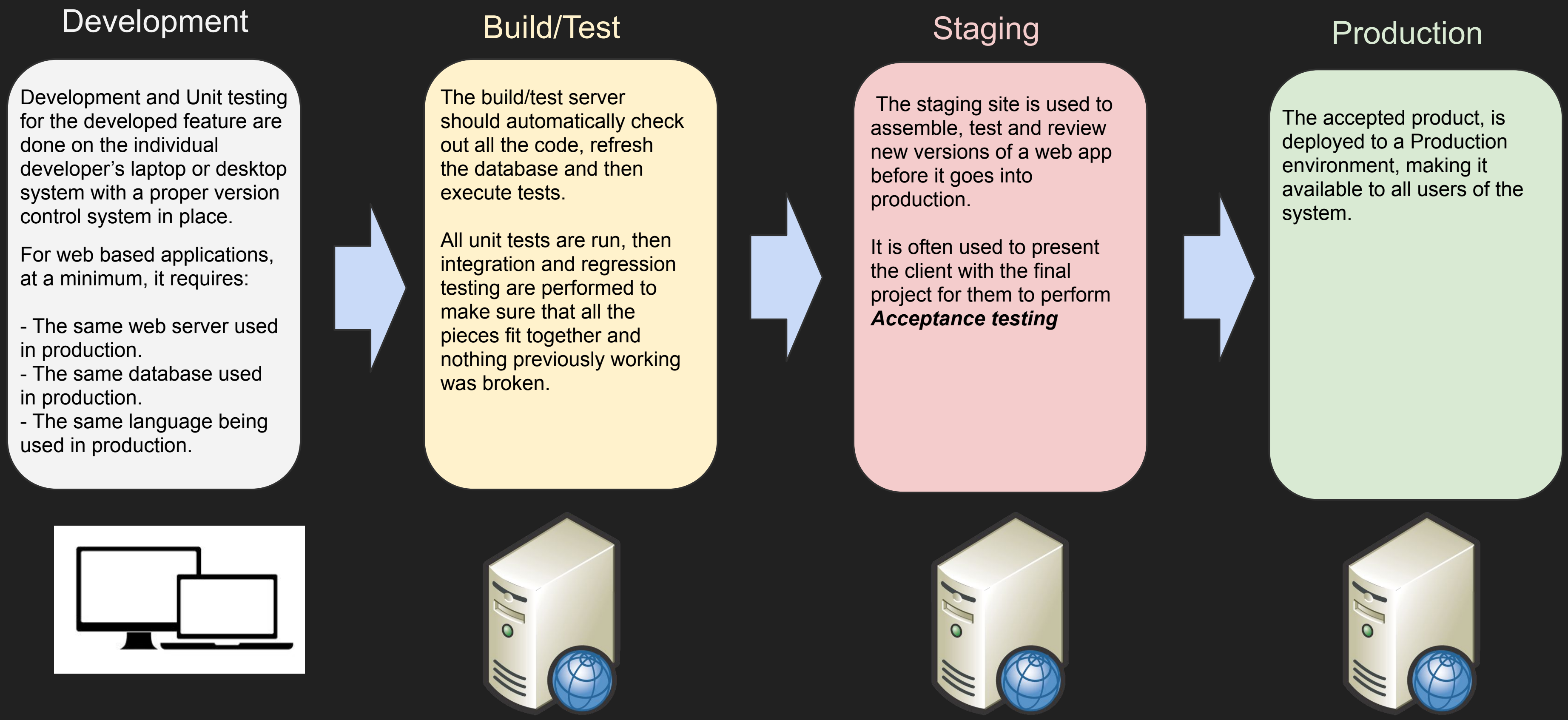
Run Test with Headless Chrome

The Product Pipeline



Environments

NOTE: Referred also as Development, Testing, Acceptance and Production (DTAP)



Development

Development and Unit testing for the developed feature are done on the individual developer’s laptop or desktop system with a proper version control system in place.

For web based applications, at a minimum, it requires:

- The same web server used in production.
- The same database used in production.
- The same language being used in production.

Build/Test

The build/test server should automatically check out all the code, refresh the database and then execute tests.

All unit tests are run, then integration and regression testing are performed to make sure that all the pieces fit together and nothing previously working was broken.

Staging

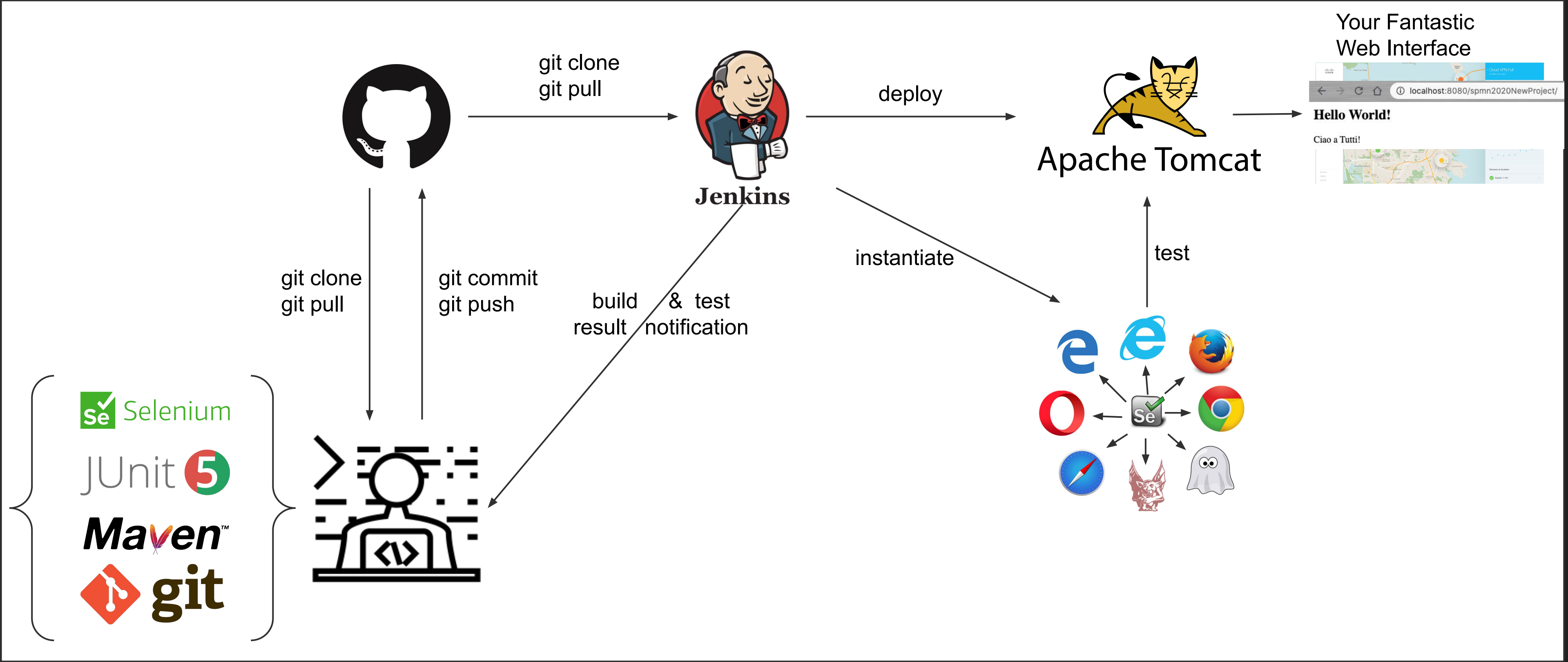
The staging site is used to assemble, test and review new versions of a web app before it goes into production.

It is often used to present the client with the final project for them to perform ***Acceptance testing***

Production

The accepted product, is deployed to a Production environment, making it available to all users of the system.

Our DevOps Toolchain



Your Fantastic Web Interface

localhost:8080/spmn2020NewProject/

Hello World!

Ciao a Tutti!

git clone
git pull

Jenkins

deploy

Apache Tomcat

git clone
git pull

git commit
git push

build & test
result notification

instantiate

test

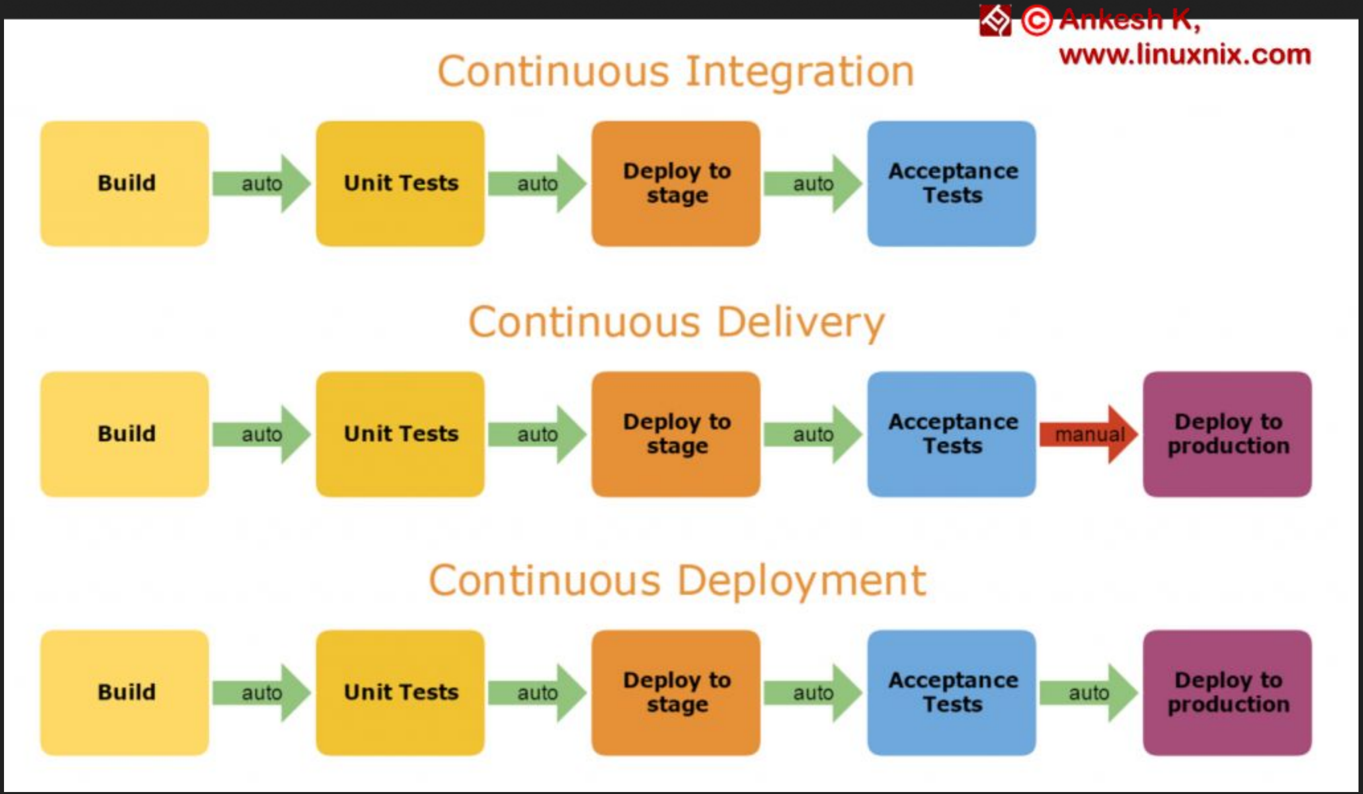
Selenium

JUnit 5

Maven

git

The Product Pipeline



© Ankesh K, www.linuxnix.com

Continuous Integration

Build → auto → Unit Tests → auto → Deploy to stage → auto → Acceptance Tests

Continuous Delivery

Build → auto → Unit Tests → auto → Deploy to stage → auto → Acceptance Tests → manual → Deploy to production

Continuous Deployment

Build → auto → Unit Tests → auto → Deploy to stage → auto → Acceptance Tests → auto → Deploy to production

Running Acceptance Tests

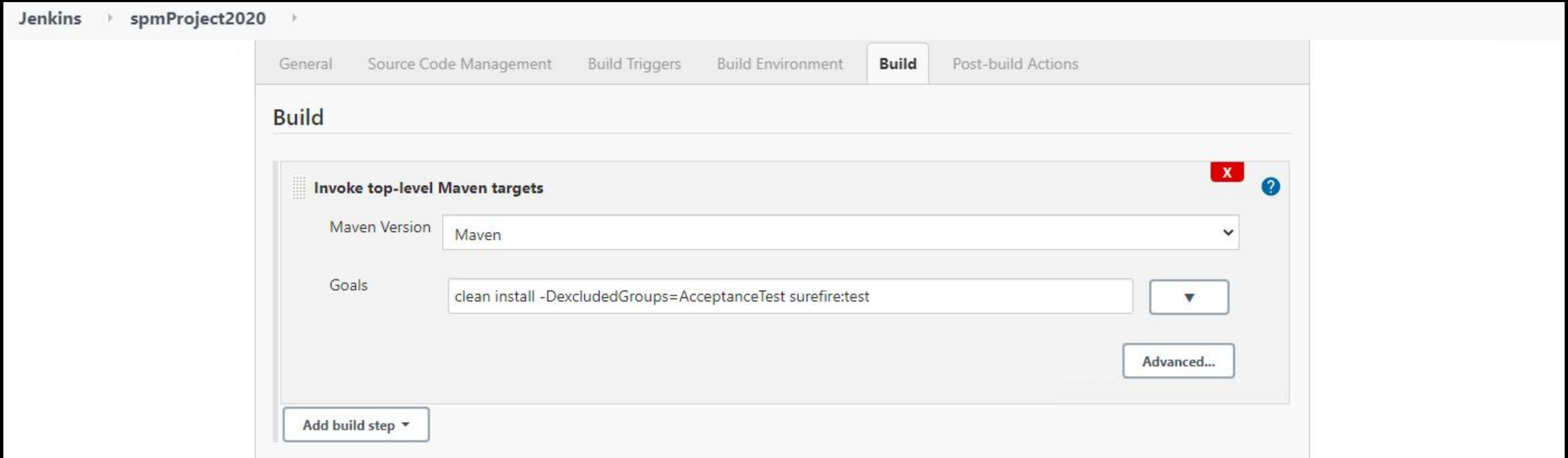
Which issue do we had last time?

A possible solution is to exclude Acceptance tests from our first Jenkins Job and include them into a second Jenkins Job

One way is to assign them a Tag and specify the maven goals:

To skip the acceptance tests: clean install -DexcludedGroups=AcceptanceTest surefire:test
To run the acceptance tests: test -Dgroups=AcceptanceTest

Modify the First Job



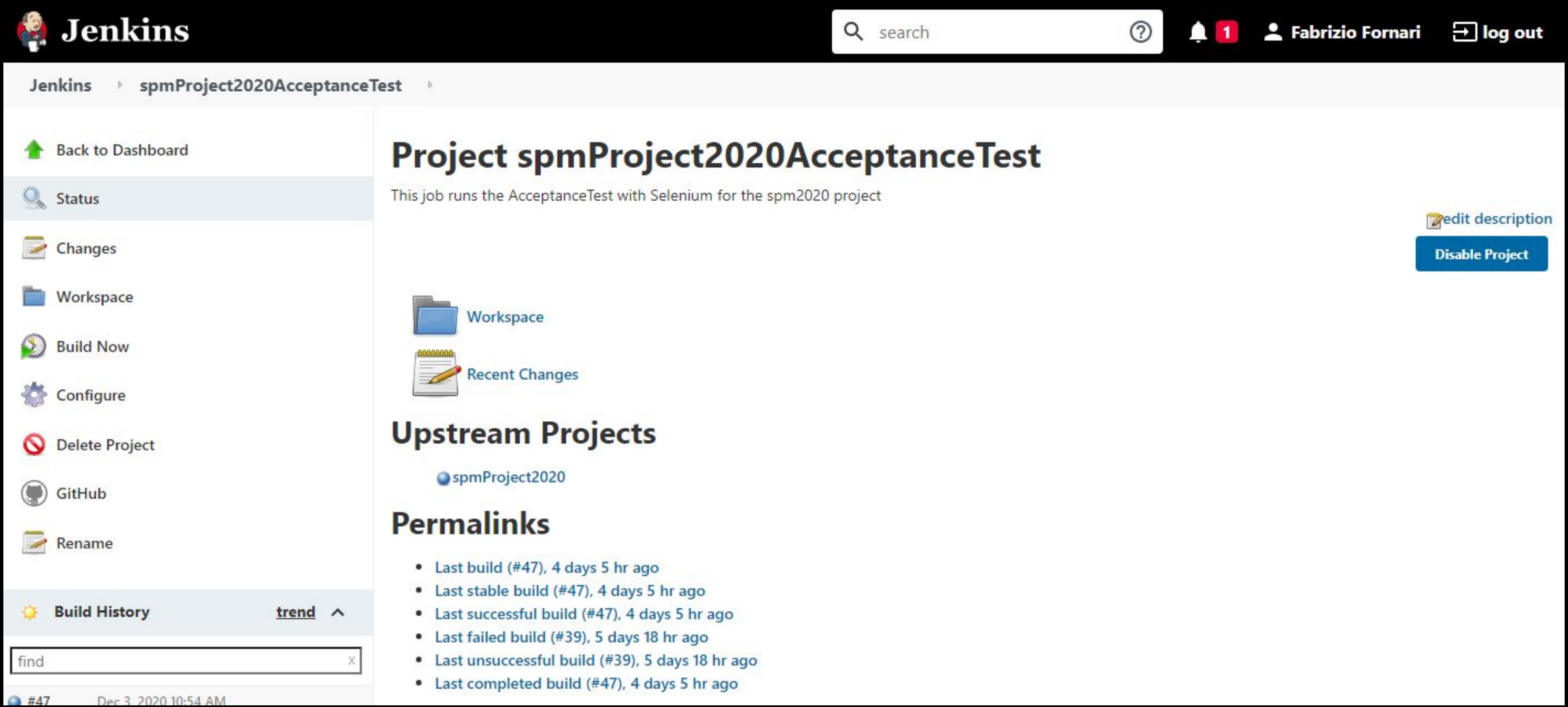
Build

Invoke top-level Maven targets

Maven Version: Maven

Goals: clean install -DexcludedGroups=AcceptanceTest surefire:test

Create the Second Job



Project spmProject2020AcceptanceTest

This job runs the AcceptanceTest with Selenium for the spm2020 project

Workspace

Recent Changes

Upstream Projects

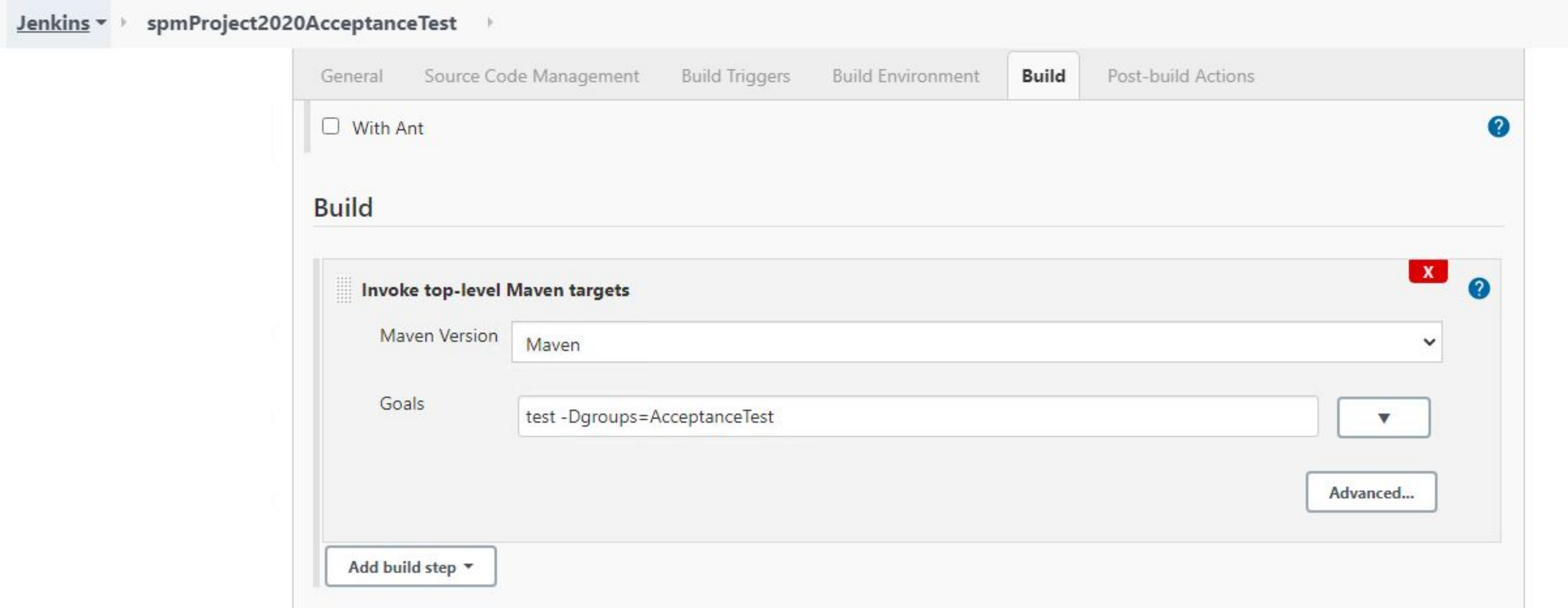
spmProject2020

Permalinks

- Last build (#47), 4 days 5 hr ago
- Last stable build (#47), 4 days 5 hr ago
- Last successful build (#47), 4 days 5 hr ago
- Last failed build (#39), 5 days 18 hr ago
- Last unsuccessful build (#39), 5 days 18 hr ago
- Last completed build (#47), 4 days 5 hr ago

#47 Dec 3, 2020 10:54 AM

Configure the Second Job



□ With Ant

Build

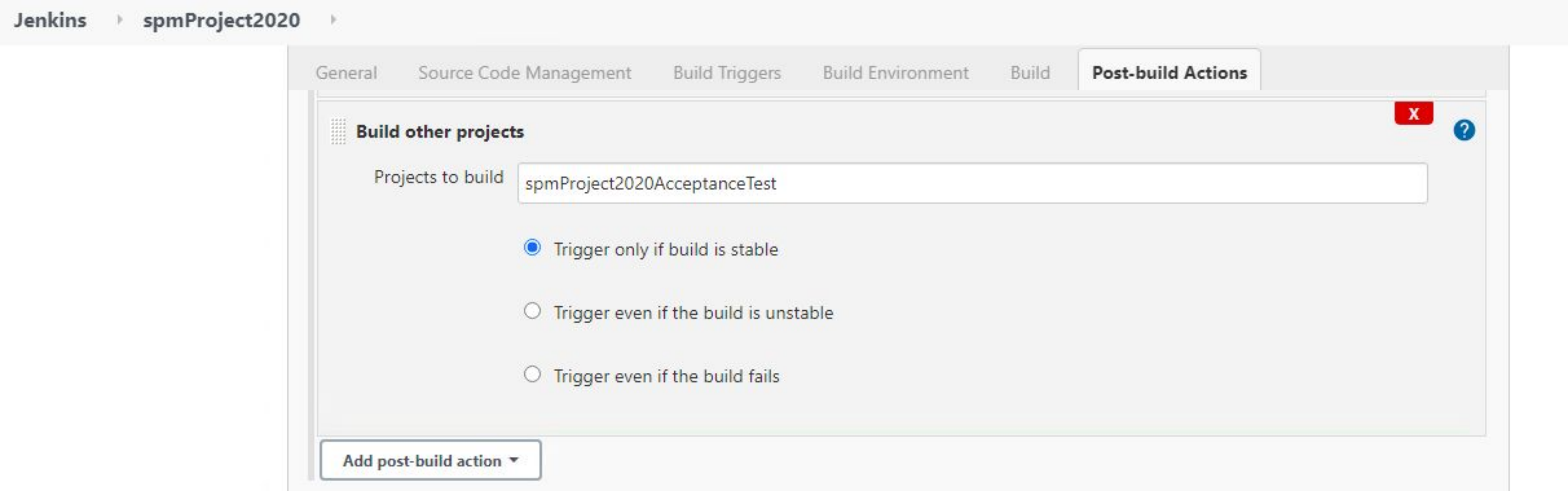
Invoke top-level Maven targets

Maven Version: Maven

Goals: test -Dgroups=AcceptanceTest

...and now?

Modify the First Job



Build other projects

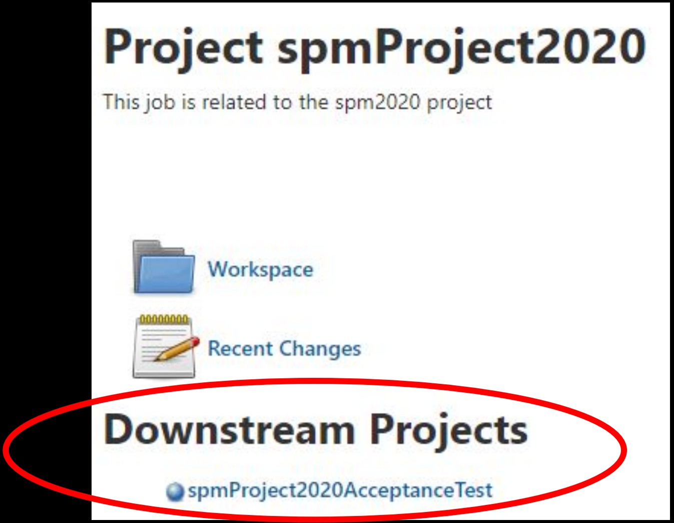
Projects to build: spmProject2020AcceptanceTest

● Trigger only if build is stable

○ Trigger even if the build is unstable

○ Trigger even if the build fails

Downstream/Upstream

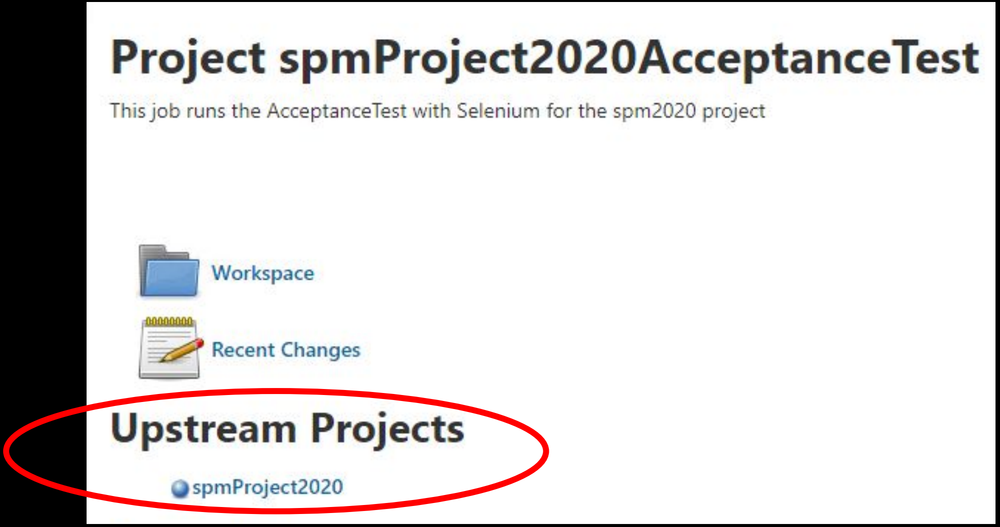


Project spmProject2020

This job is related to the spm2020 project

Downstream Projects

spmProject2020AcceptanceTest



Project spmProject2020AcceptanceTest

This job runs the AcceptanceTest with Selenium for the spm2020 project

Upstream Projects

spmProject2020

Jenkins Plugins

https://plugins.jenkins.io/

Not only Jenkins

TRAVIS CI

https://travis-ci.org/

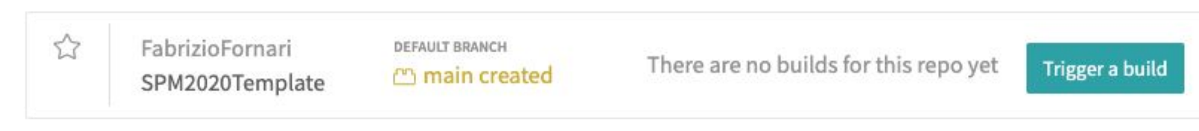
Travis CI

Travis CI is a hosted, distributed continuous integration service used to build and test software projects hosted at GitHub.

Travis CI is configured by adding a file named .travis.yml, which is a YAML format text file, to the root directory of the repository.

```
# this is a java project using maven
language: java
# install
install: mvn install
```

https://travis-ci.org/



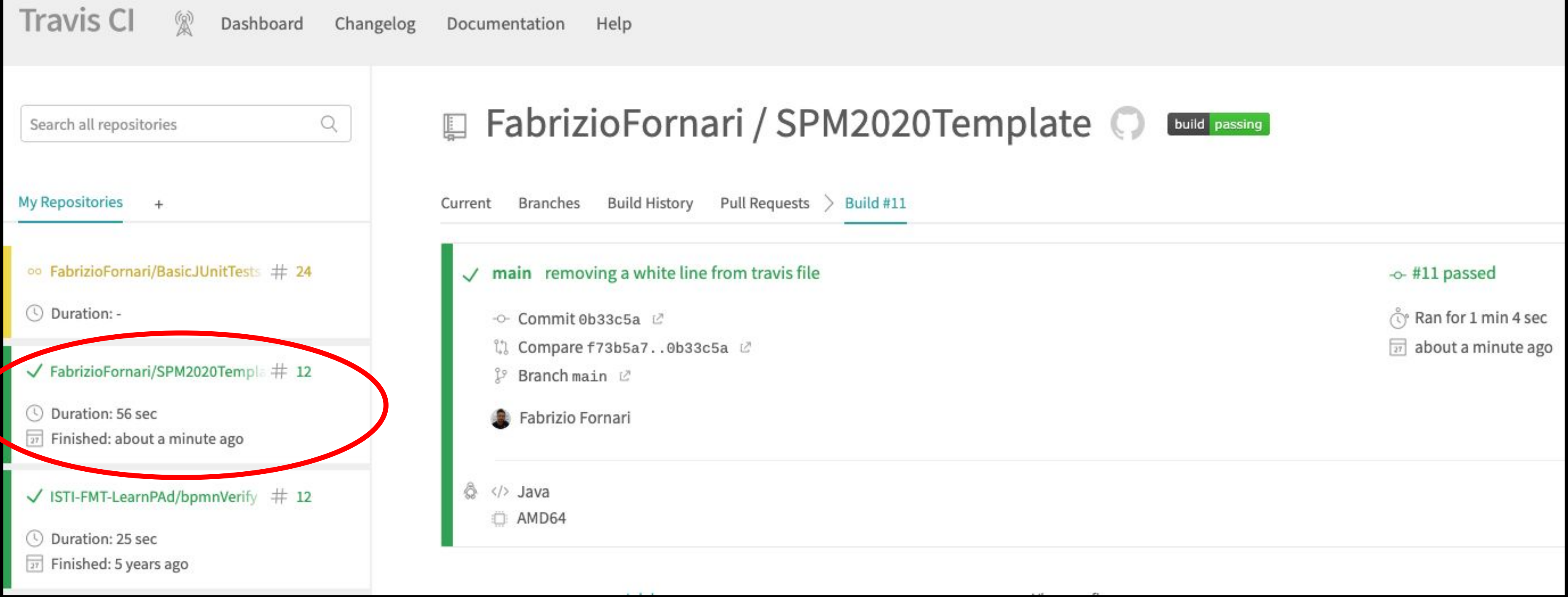
FabrizioFornari
SPM2020Template

DEFAULT BRANCH
main created

There are no builds for this repo yet

Travis CI Documentation: https://docs.travis-ci.com/

Travis CI



FabrizioFornari/BasicJUnitTests # 24
Duration: -

✓ FabrizioFornari/SPM2020Templa # 12
Duration: 56 sec
Finished: about a minute ago

✓ ISTI-FMT-LearnPAd/bpmnVerify # 12
Duration: 25 sec
Finished: 5 years ago

FabrizioFornari / SPM2020Template

✓ main removing a white line from travis file

#11 passed

Commit 0b33c5a
Compare f73b5a7..0b33c5a
Branch main

Ran for 1 min 4 sec
about a minute ago

Fabrizio Fornari

Java
AMD64

Recap

[Project Evaluation Schema]