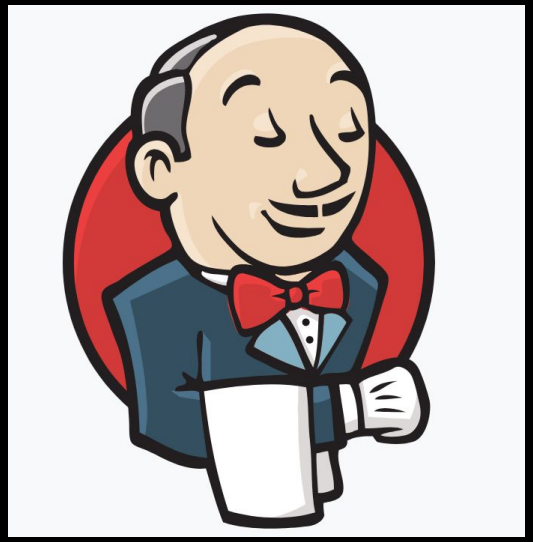
Software Project Management - Laboratory

Lecture n° 13
A.Y. 2021-2022

Prof. Fabrizio Fornari

Jenkins

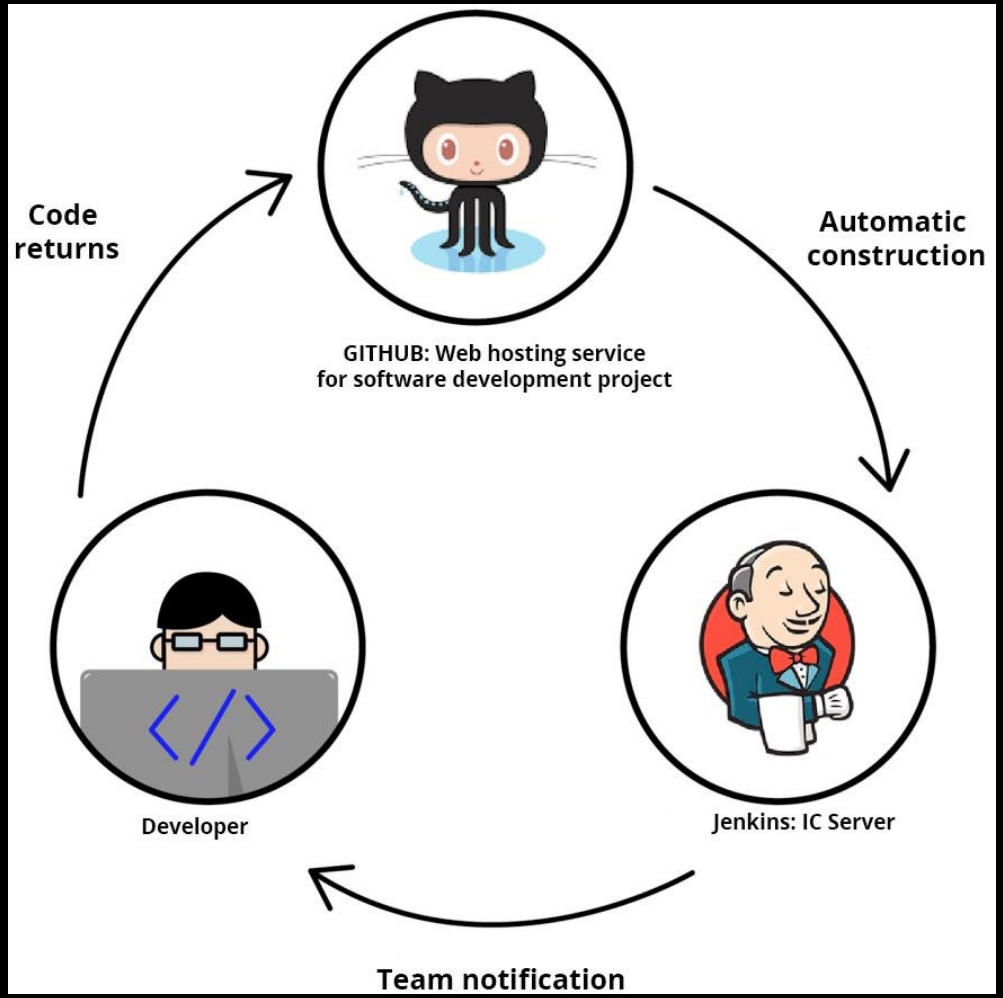


Jenkins is used to build and test your product continuously, so developers can continuously integrate changes into the build.

https://jenkins.io/

Continuous Integration with Jenkins

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



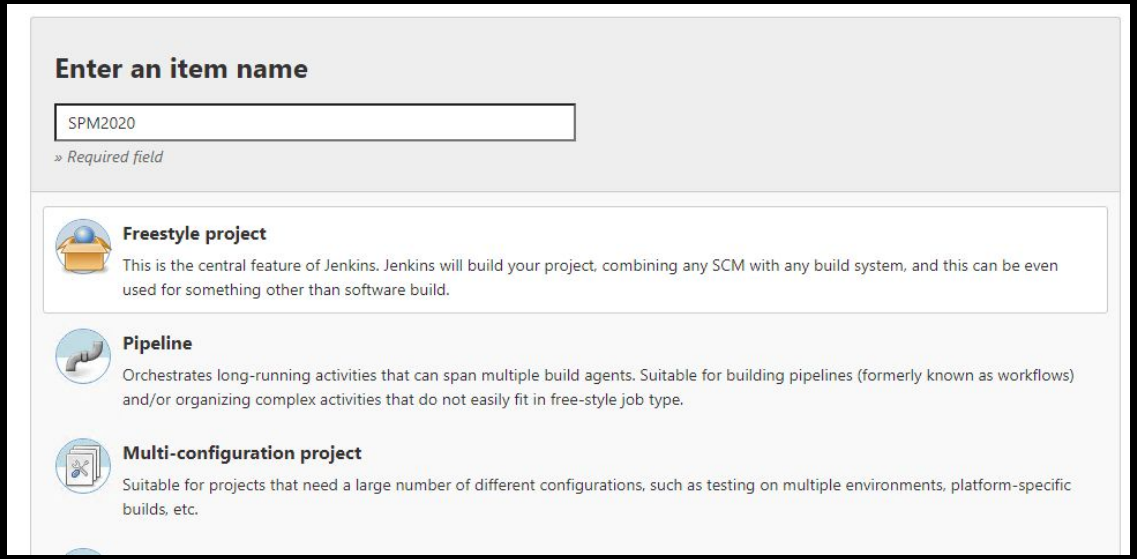
Jenkins on Tomcat

Download the latest .war file
https://www.jenkins.io/download/

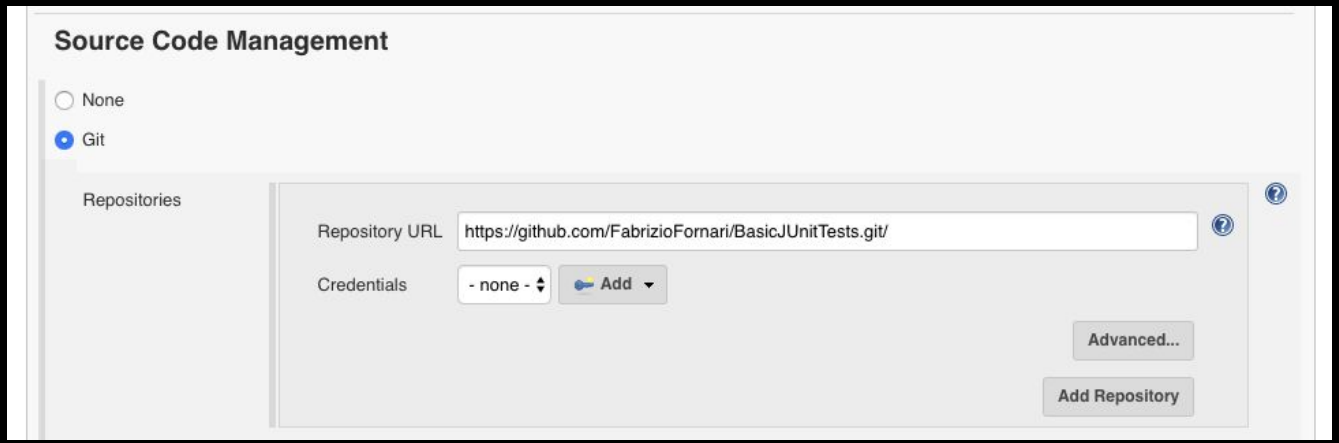
Deploy the .war on Tomcat
https://www.jenkins.io/doc/book/installing/#war-file

Create Job

1 Create a new Job

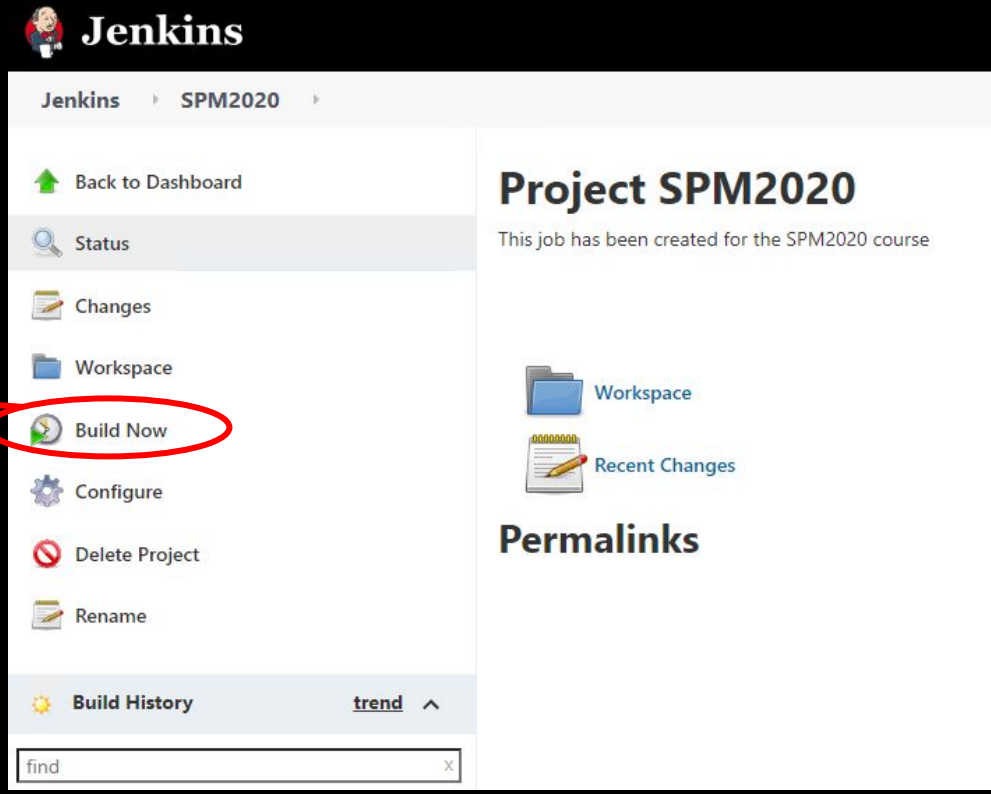


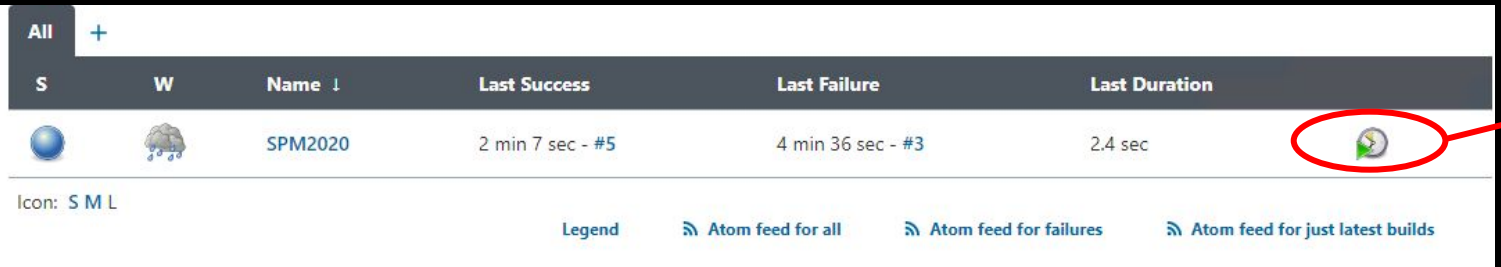
2 Link it with your GitHub Repository



Run a Job

Run a Job

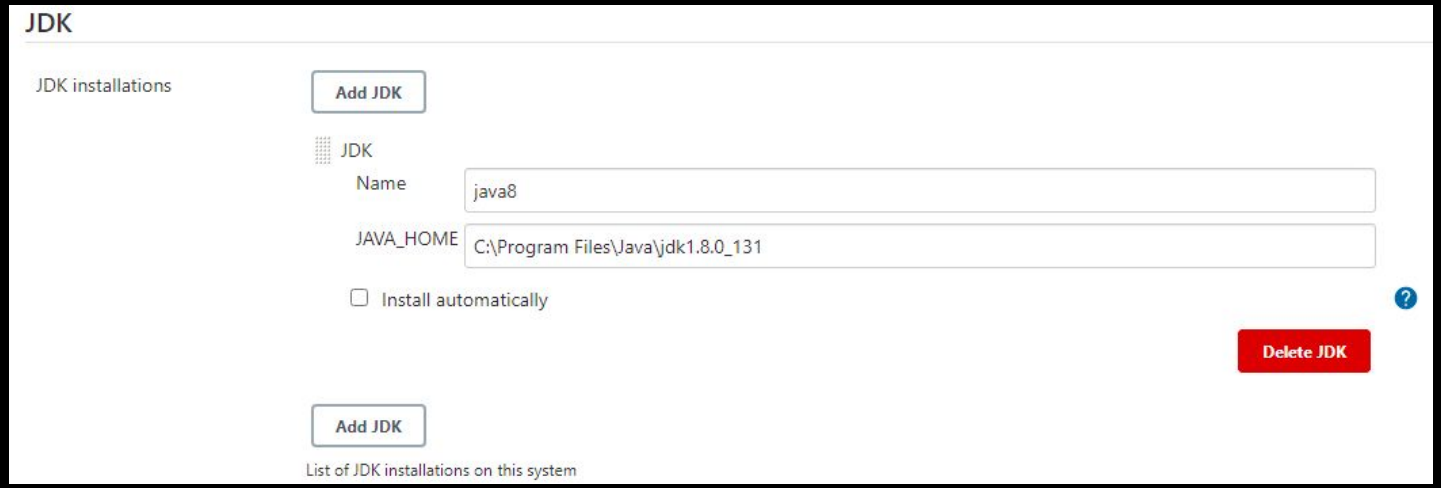




Run a Job

Configure JDK

Manage Jenkins > Global Tool Configuration

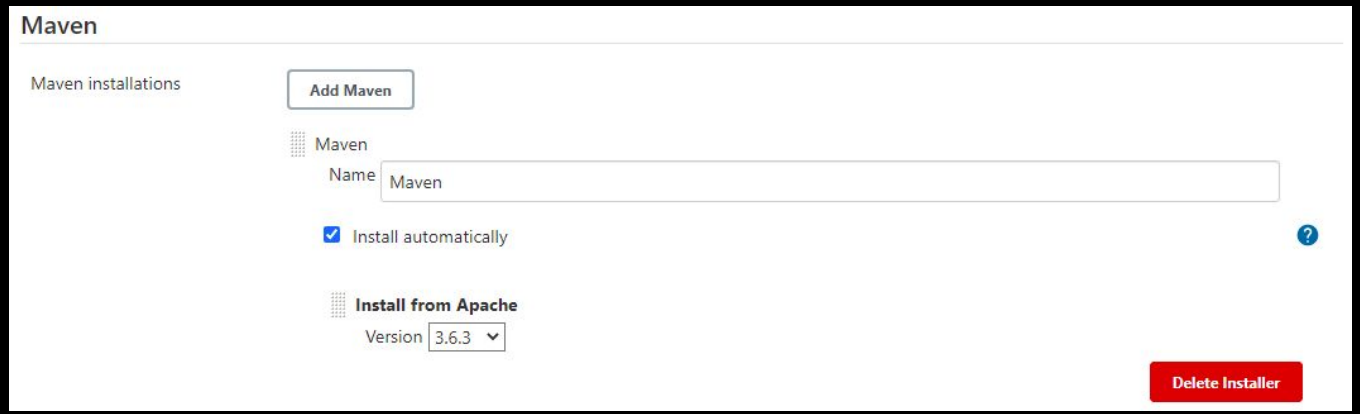


Configure JDK & Maven

Manage Jenkins --> Global Tool Configuration

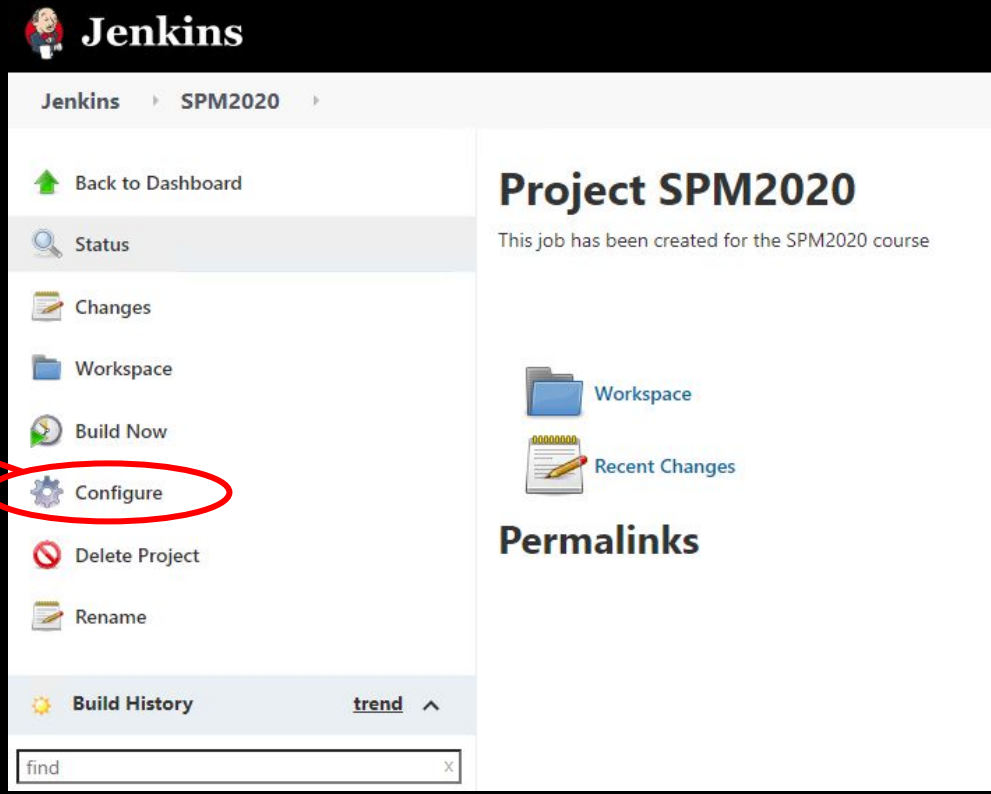
Or

Tell Jenkins to install a version



Configure a Job

Configure for
changing settings



Configure Build Environment

Schedule a build

(Build Triggers → Build periodically)

Jenkins uses a cron expression, and the different fields are:

1. MINUTES Minutes in one hour (0-59)
2. HOURS Hours in one day (0-23)
3. DAYMONTH Day in a month (1-31)
4. MONTH Month in a year (1-12)
5. DAYWEEK Day of the week (0-7) where 0 and 7 are sunday

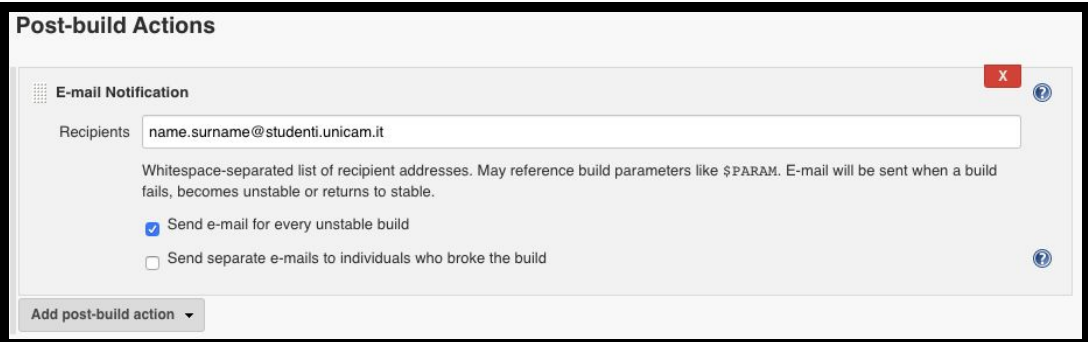
MINUTES	HOURS	DAYMONTH	MONTH	DAYWEEK
*/5	*	*	*	*

To schedule a build every 5 minutes : */5 * * * *
To schedule a build every day at 8h00, this will do the job : 0 8 * * *

Email Notifications

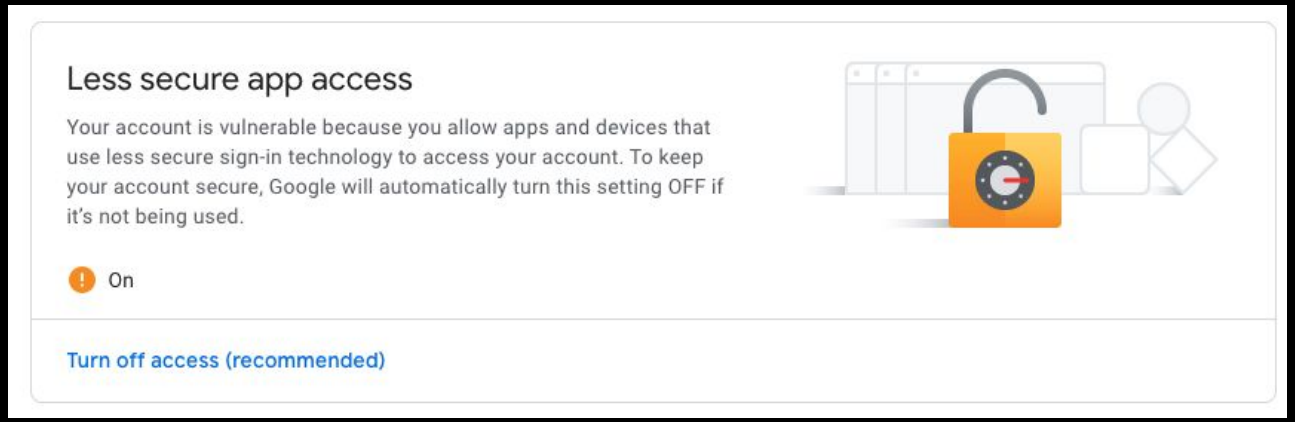
Manage Jenkins → Configure System

Job Name --> Configure



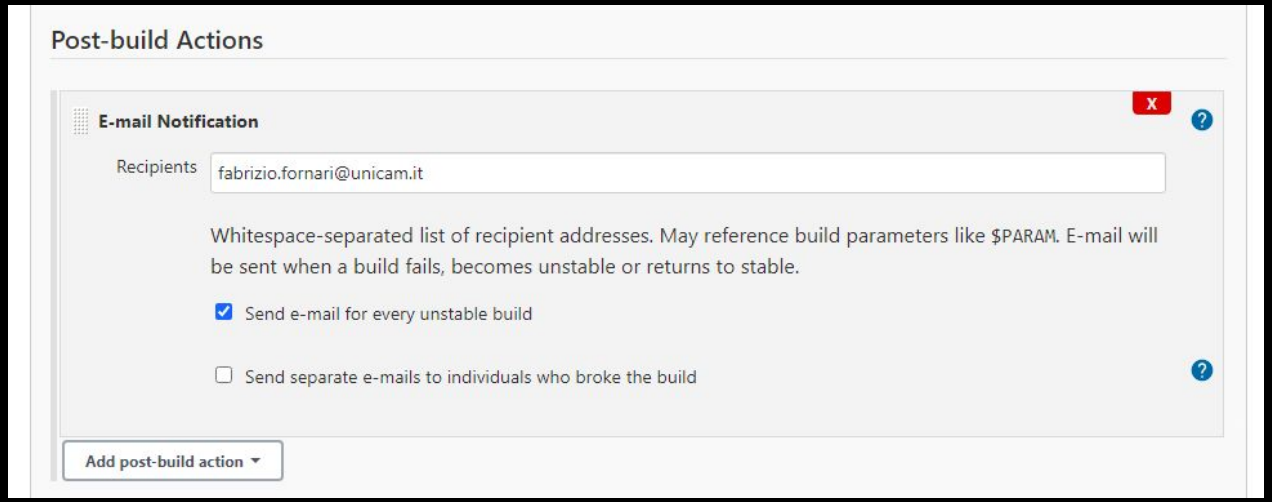
Email Notifications

If you are using gmail, allow third party application to login from https://myaccount.google.com/security



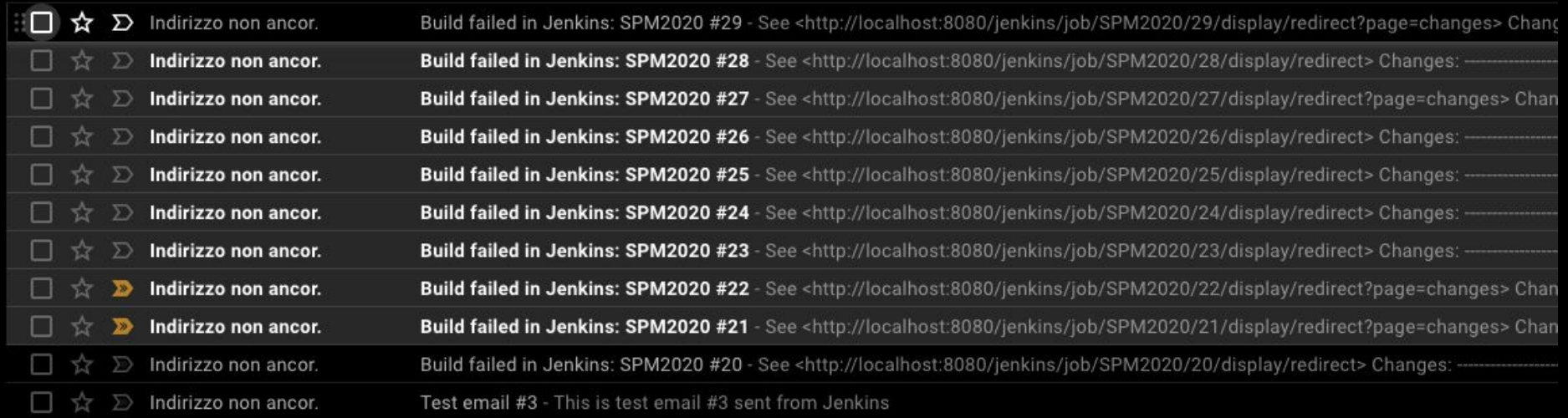
Test the configuration

Email Notifications



Test the configuration

Email Notifications



To be removed: https://mail.google.com/mail/u/0/#inbox/FMfcgxwKjdtDKSXnZkfdzMGGxxBRtLKj

Build failed in Jenkins: SPM2021

See <http://proslabtest:8080/jenkins/job/SPM2021/4/display/redirect?page=changes>
Changes:
[fabrizio.fornari] added a failing test

Started by user Fabrizio Fornari
Running as SYSTEM
Building in workspace <http://proslabtest:8080/jenkins/job/SPM2021/ws/>
…
[INFO] ----------------------< pros.unicam.it:spm2021 >----------------------
[INFO] Building webapp Maven Webapp 0.0.1-SNAPSHOT
[INFO] --------------------------------[war]---------------------------------
[INFO] --- maven-clean-plugin:3.1.0:clean (default-clean) @ spm2021 ---
[INFO] Deleting <http://proslabtest:8080/jenkins/job/SPM2021/ws/target>
…
[INFO] --- maven-compiler-plugin:3.8.0:testCompile (default-testCompile) @ spm2021 ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 7 source files to <http://proslabtest:8080/jenkins/job/SPM2021/ws/target/test-classes>
...

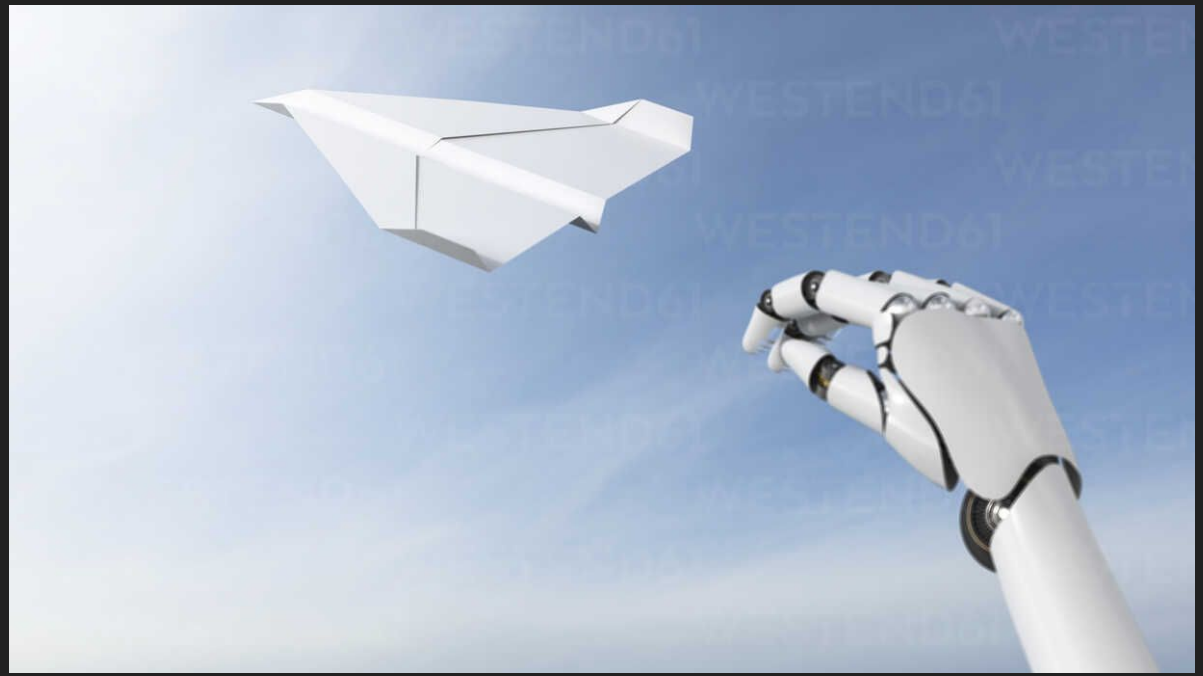
...
[INFO] ---
[INFO] T E S T S
[INFO] ---
[INFO] Running pros.unicam.spm2021.practice.JUnit.EmptyTest
[WARNING] Tests run: 1, Failures: 0, Errors: 0, Skipped: 1, Time elapsed: 0.05 s - in pros.unicam.spm2021.practice.JUnit.EmptyTest
[INFO] Running pros.unicam.spm2021.practice.JUnit.HelloWorldTest
Nov 22, 2021 11:54:31 AM pros.unicam.spm2021.practice.JUnit.HelloWorldTest setUpBeforeClass
[ERROR] Tests run: 12, Failures: 1, Errors: 0, Skipped: 3, Time elapsed: 0.185 s <<< FAILURE! - in pros.unicam.spm2021.practice.JUnit.HelloWorldTest
[ERROR] testMain Time elapsed: 0.019 s <<< FAILURE!
org.opentest4j.AssertionFailedError: Not yet implemented
	at pros.unicam.spm2021.practice.JUnit.HelloWorldTest.testMain(HelloWorldTest.java:111)
[INFO]
[INFO] --
[INFO] BUILD FAILURE
[INFO] --
[INFO] Total time: 9.285 s
[INFO] Finished at: 2021-11-22T11:54:32+01:00
[INFO] --

Fix the error

Email: Jenkins build is back to normal : SPM2021 #5

See <http://proslabtest:8080/jenkins/job/SPM2021/5/display/redirect?page=changes>

Automatic Deploy



Apache Maven

Apache Maven is an open source, standards-based project management framework that simplifies the building, testing, reporting, and packaging of projects.

http://maven.apache.org/

Apache Tomcat

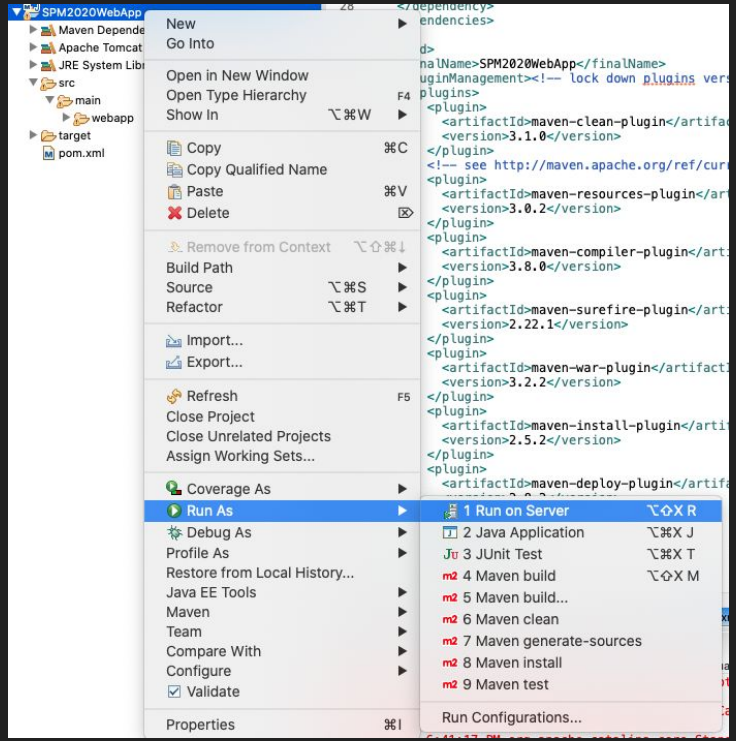
The Apache Tomcat® software is an open source implementation of the Java Servlet, JavaServer Pages, Java Expression Language and Java WebSocket technologies.

http://tomcat.apache.org/

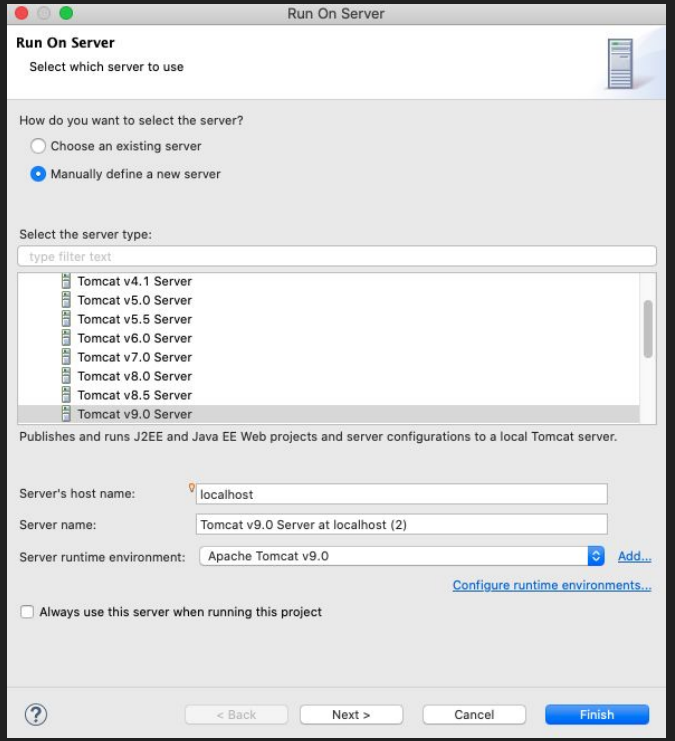
Download Tomcat
https://tomcat.apache.org/download-90.cgi

Run Your Application

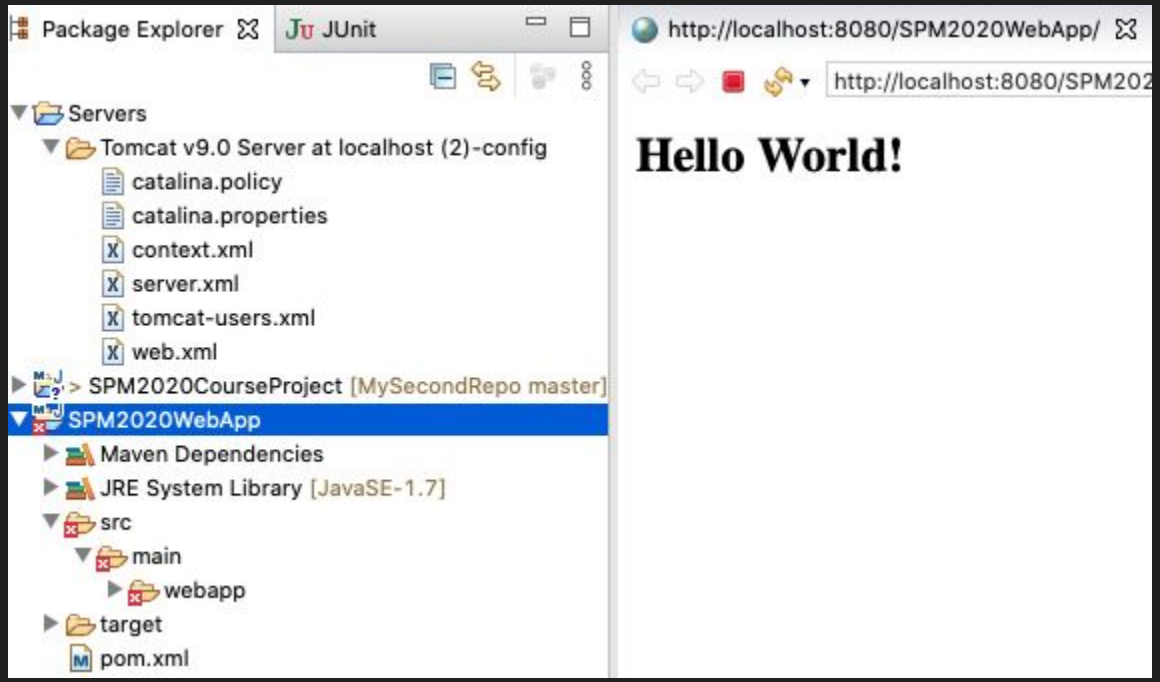
Run On Server



Pick the version you installed

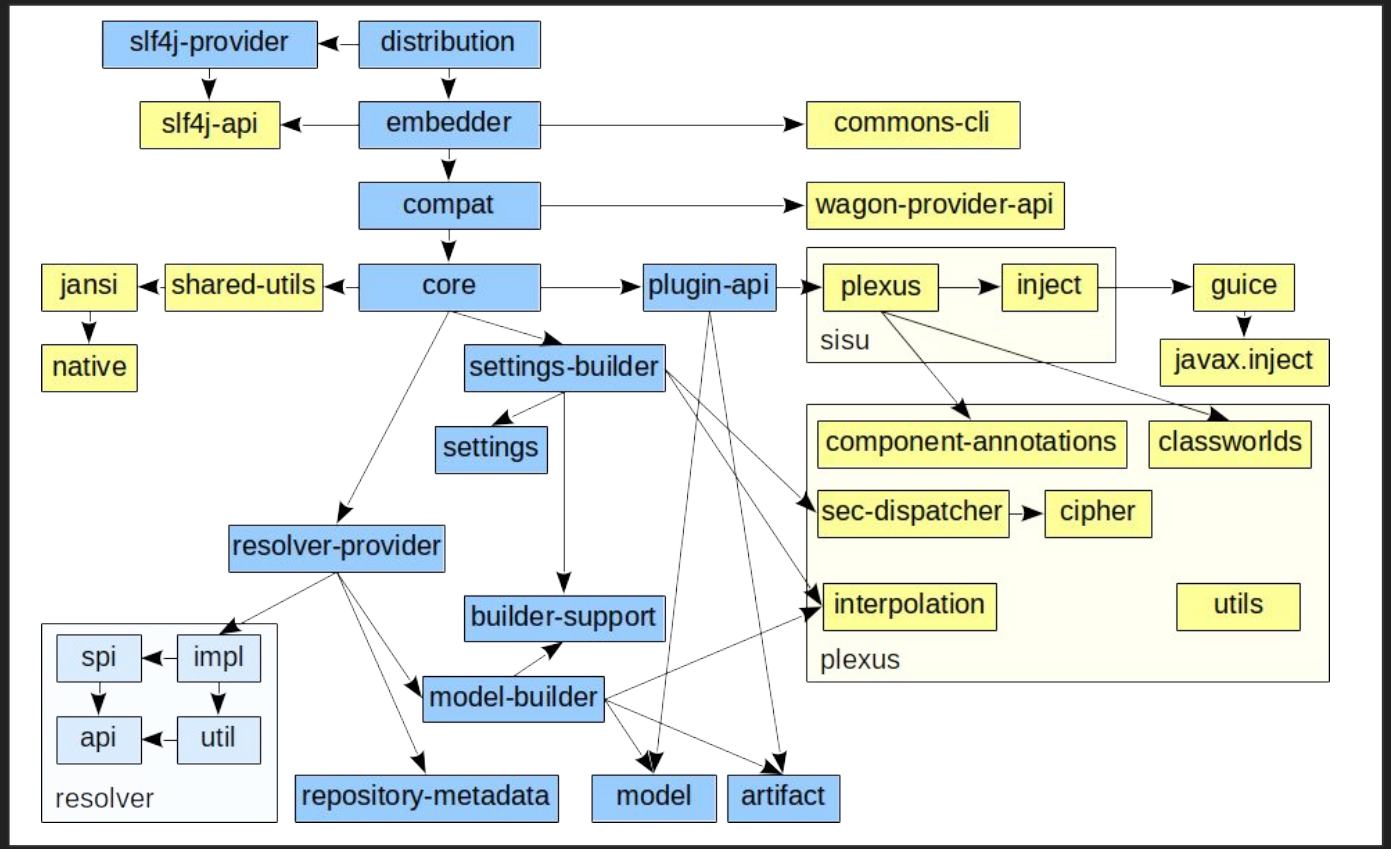


Run Your Application



Maven

Maven can be extended by plugins to utilise a number of other development tools for reporting or the build process



Automatically deploy a WAR on Tomcat with Maven

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="maven-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.

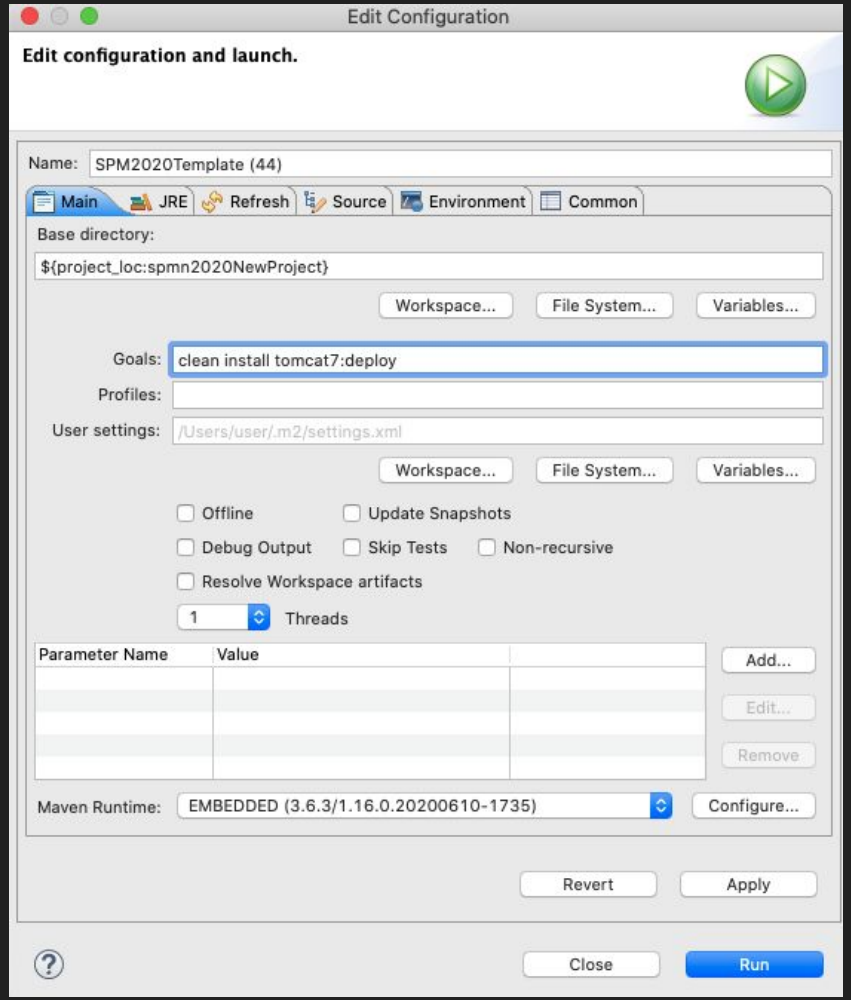
Automatically deploy a WAR on Tomcat with Maven

```
<!-- Configure the Tomcat Maven plugin -->
                <plugin>
                        <groupId>org.apache.tomcat.maven</groupId>
                        <artifactId>tomcat7-maven-plugin</artifactId>
                        <version>2.2</version>
                        <configuration>
                                <!-- Use tomcat9 user defined credentials.
                                Usually you would place them under the Maven
                                folder .settings.xml telling eclipse to read them from there
                                by just using adding <server>tomcat9</server>-->
                                <username>war-deployer</username>
                                <password>maven-tomcat-plugin</password>
                                <update>true</update>
                                <!-- <url>http://localhost:8080/manager/text</url>-->
                        </configuration>
                </plugin>
```

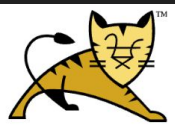
Apache Tomcat Maven Plugin: http://tomcat.apache.org/maven-plugin-2.2/

Maven Goals

Goals: clean install tomcat7:deploy



Automatically Deployed WebApp



Tomcat Web Application Manager

Message:	OK

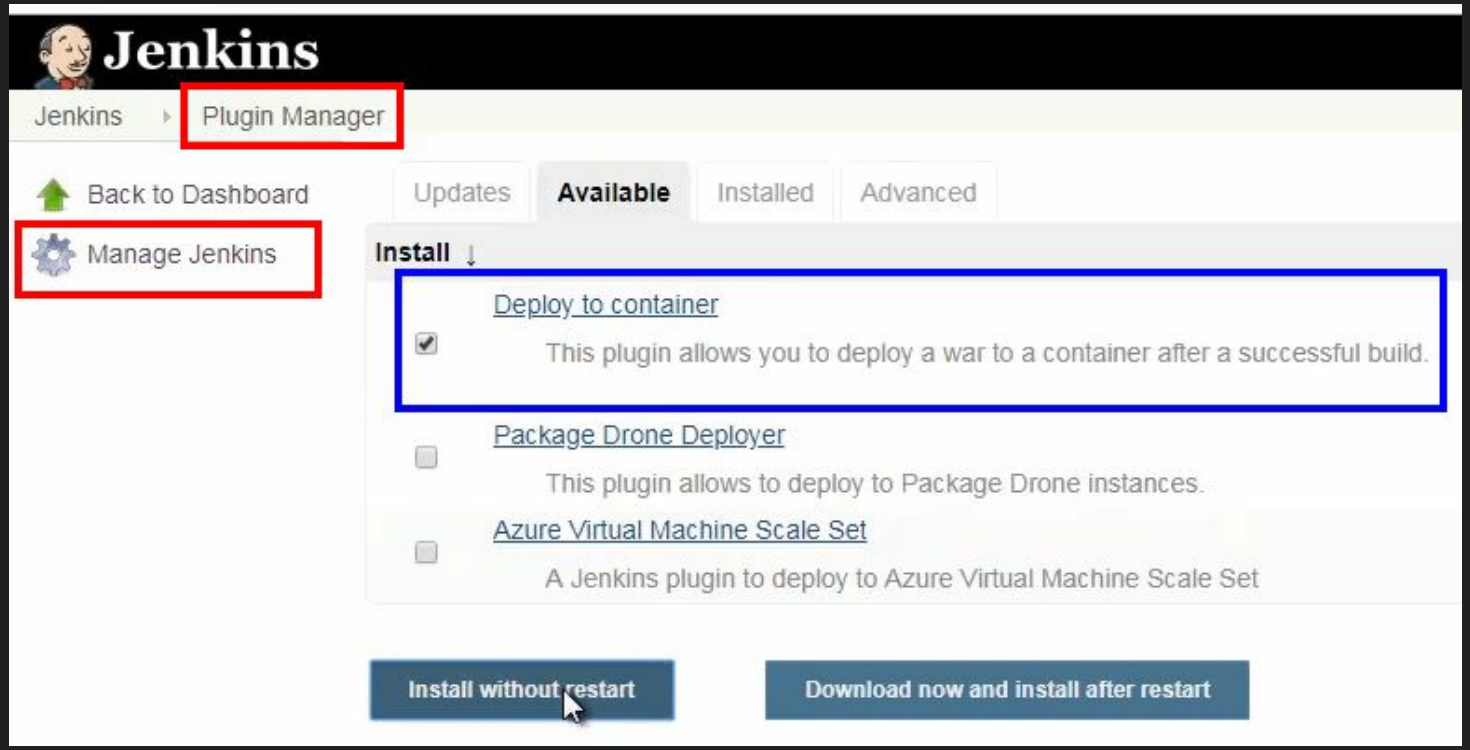
Manager			
List Applications	HTML Manager Help	Manager Help	Server Status

Applications

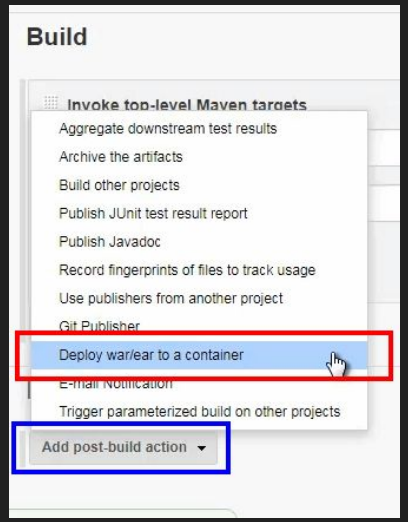
Path	Version	Display Name	Running	Sessions	Commands
/	*None specified*	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/docs	*None specified*	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/examples	*None specified*	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 v2.321	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/manager	*None specified*	Tomcat Manager Application	true	2	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/spm2021	*None specified*	Archetype Created Web Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

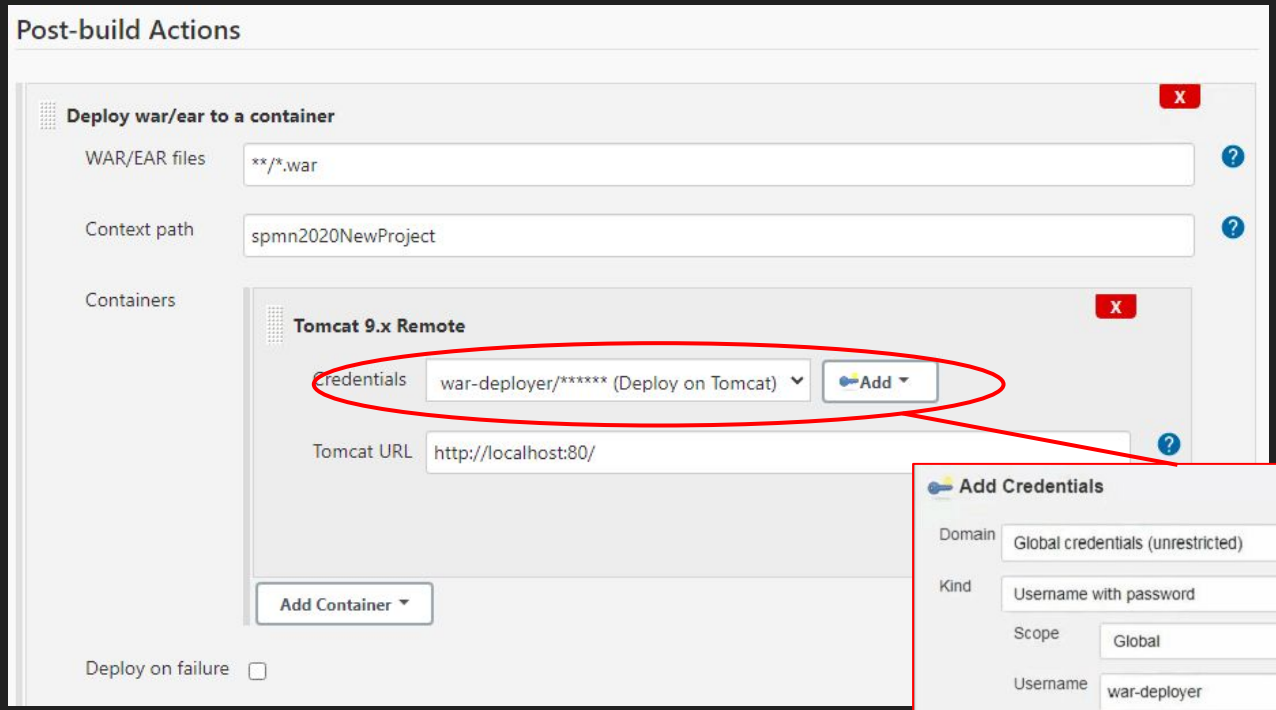
Automatically deploy a WAR on Tomcat with Jenkins

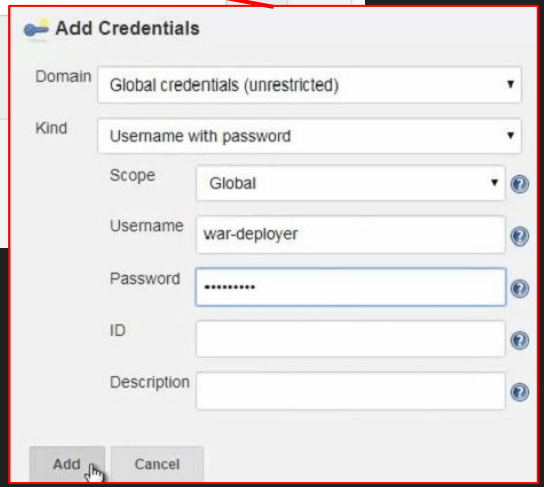
Install the Deploy to container plugin



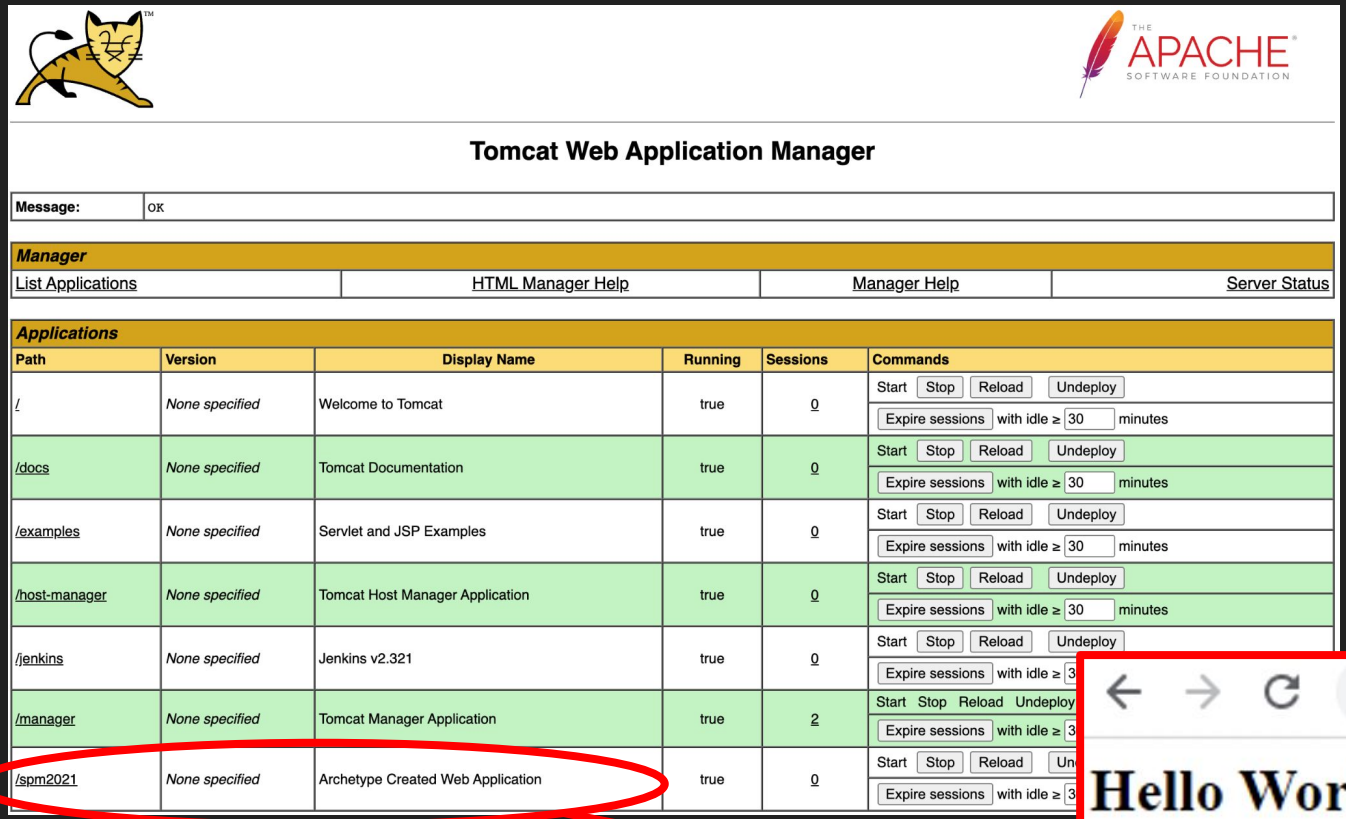
Automatically deploy a WAR on Tomcat with Jenkins







Automatically deploy a WAR on Tomcat with Jenkins



Tomcat Web Application Manager

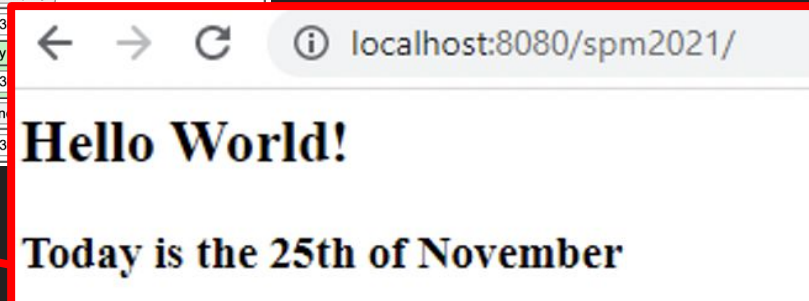
Message:	OK

Manager

List Applications	HTML Manager Help	Manager Help	Server Status

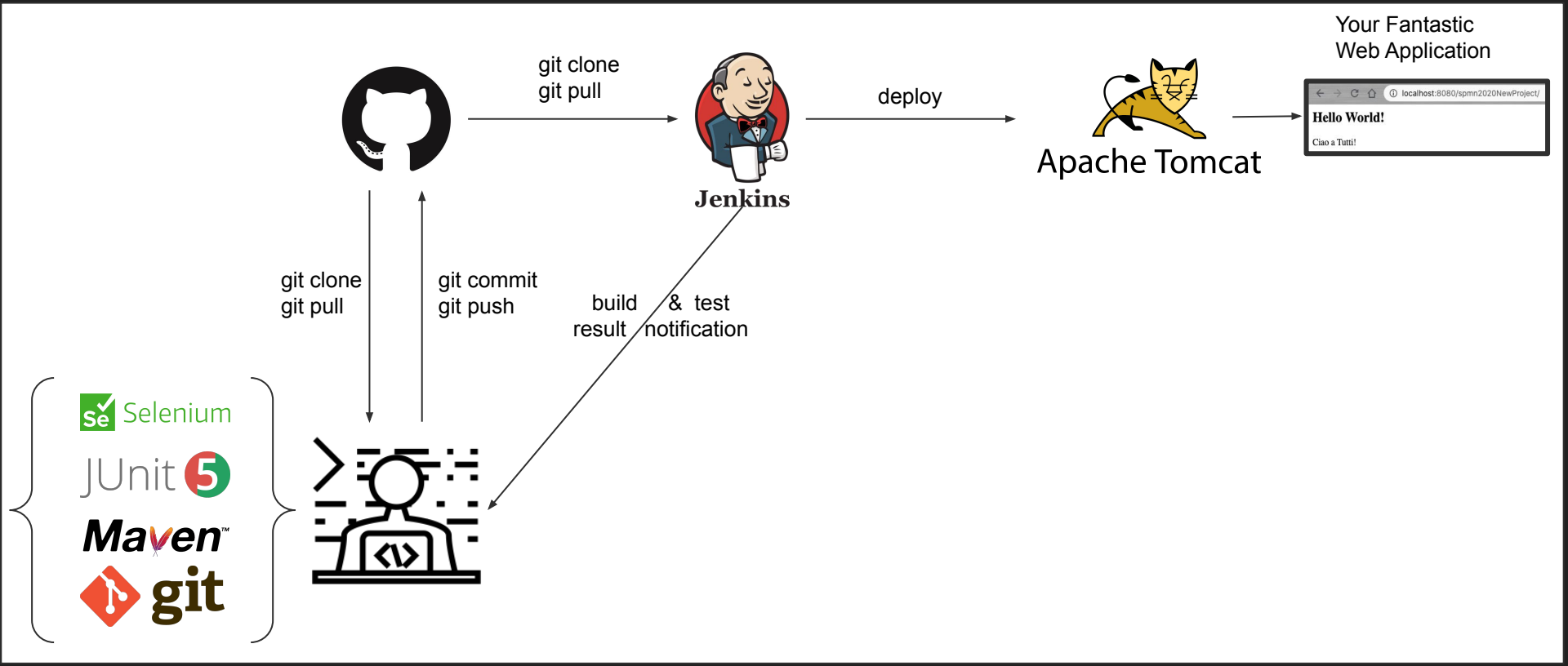
Applications

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 v2.321	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 3
/manager	*None specified*	Tomcat Manager Application	true	2	Start Stop Reload Undeploy Expire sessions with idle ≥ 3
/spm2021	*None specified*	Archetype Created Web Application	true	0	Start Stop Reload Un Expire sessions with idle ≥ 3

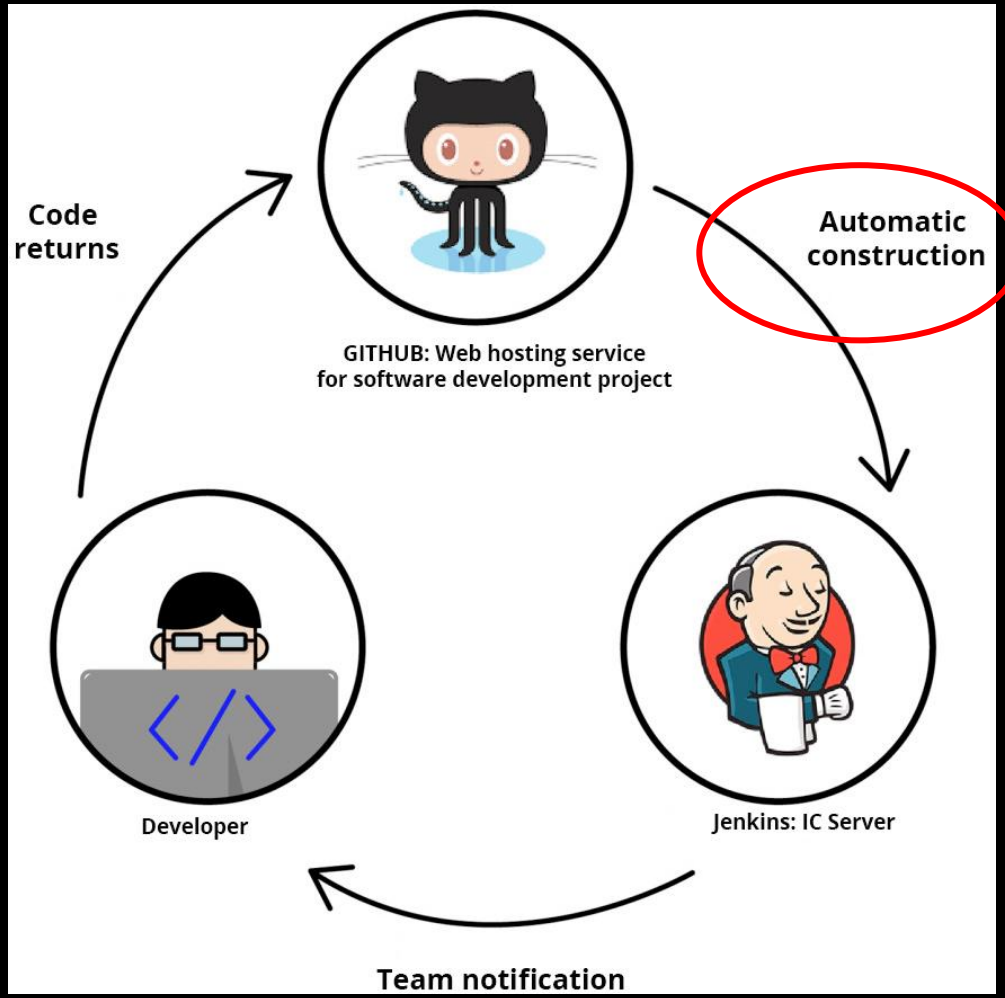


To Do: Change index.jsp

What is Missing?



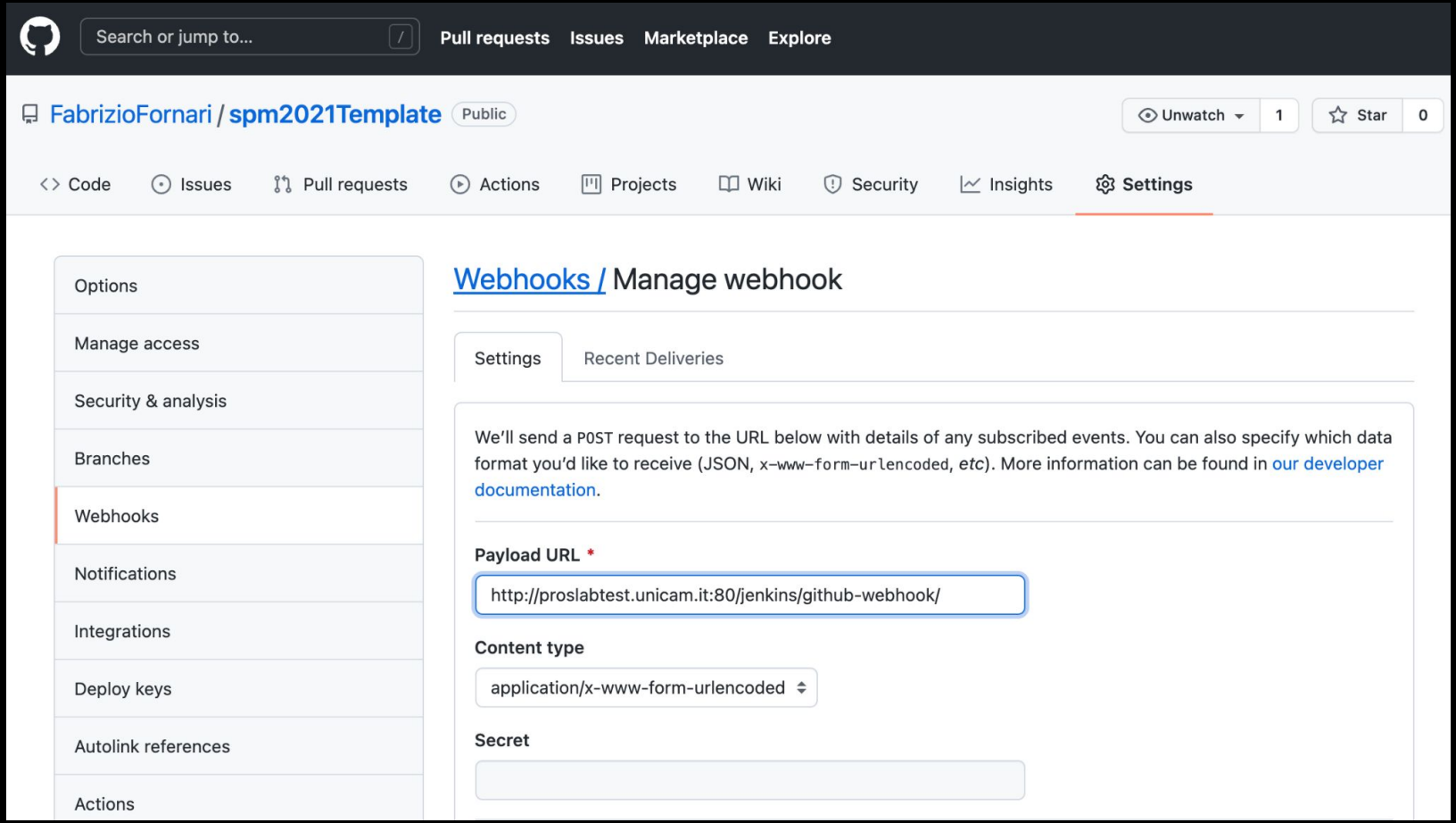
How to connect Github and Jenkins?



Webhook

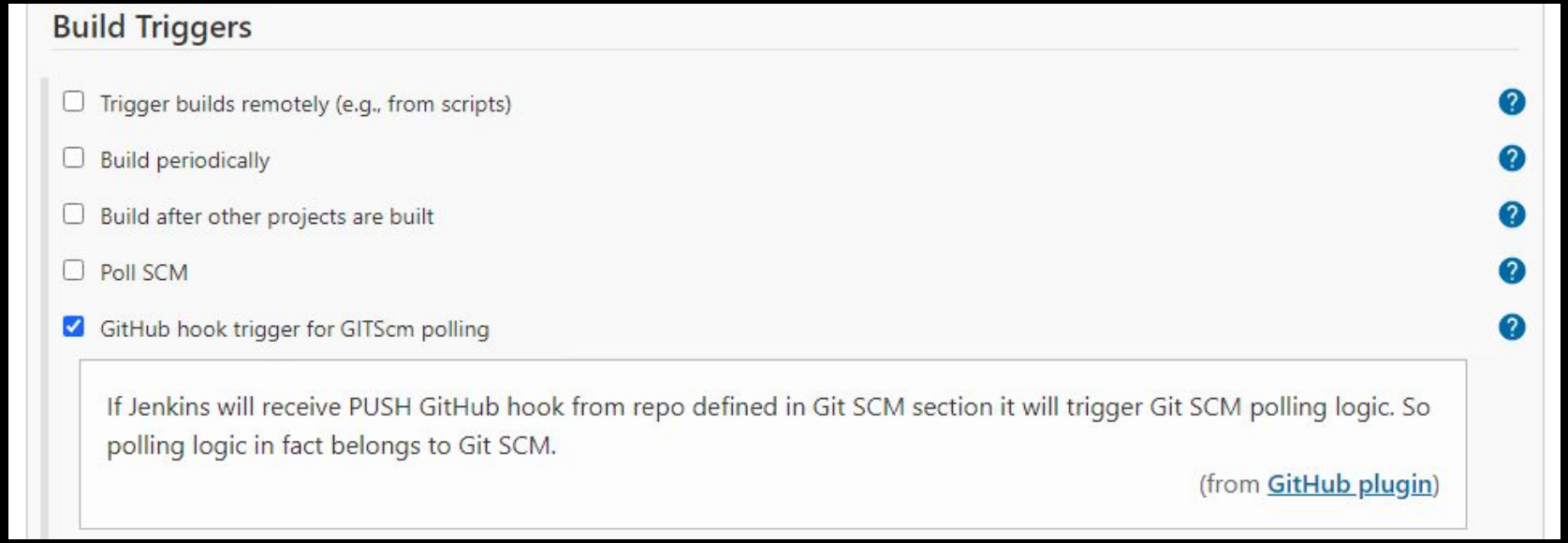
- Webhooks allow you to build or set up GitHub Apps which subscribe to certain events on GitHub.com.
- When one of those events is triggered, we'll send a HTTP POST payload to the webhook's configured URL.
- Webhooks can be used to update an external issue tracker, trigger CI builds, update a backup mirror, or even deploy to your production server.

https://developer.github.com/webhooks/



http://proslabtest.unicam.it:80/jenkins/github-webhook/

Jenkins GitHub hook trigger



Enable Selenium Tests

Is everything ok?

Environments

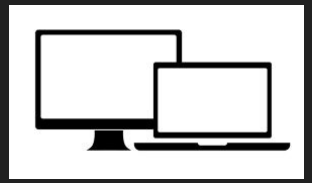
NOTE: Referred also as Development, Testing, Acceptance and Production (DTAP)

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.

The Product Pipeline