# Jenkins ePortfolio

in

Software-Engineering

from

Michaela Fleig

on date

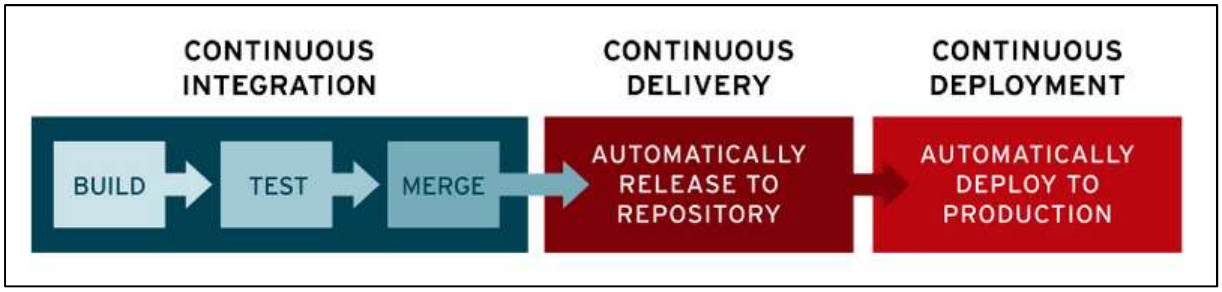
20 May 2020

# Agenda

# 1. Continuous Integration

## 1.1. Continuous Integration as part of CI/CD

Continuous Integration/Continuous Development/Continuous Deployment (CI/CD) describes the way of the application from its development to releasing. In many cases developer work together and simultaneously at an application, but each on his own part. Using CI/CD means saving the new source code on a repository. Then, others can work with this source code that can be offered to customers.




Picture 1: The phases of Continuous Integration, -Delivery and –Deployment

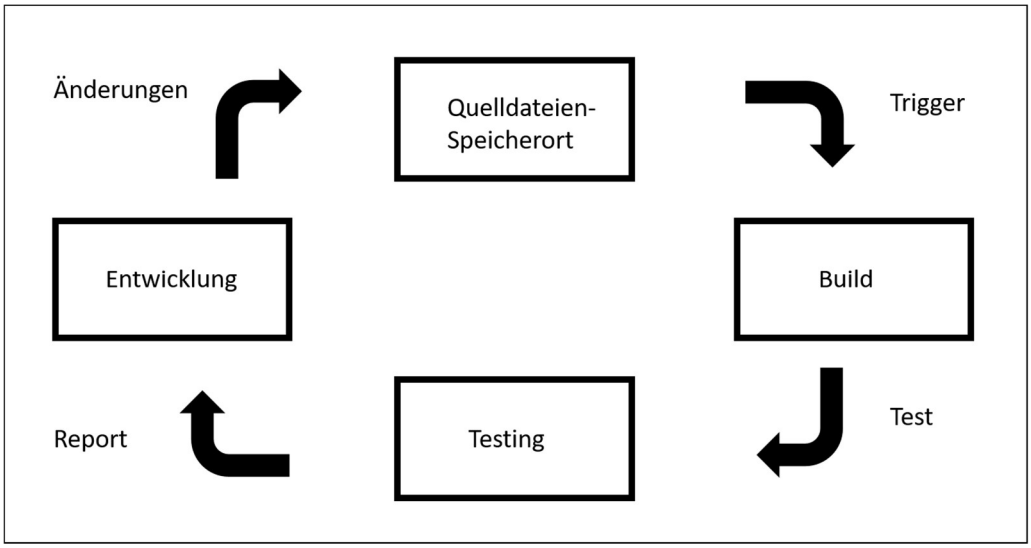
Picture 1 shows the three parts of the Continuous Integration “build”, “test” and “merge”. At “build” most syntax errors are found. The tests are in the means of agile development and have to be written extra for the source code. When merging, the code is stored in a staging system. When the test fails, the developer is notified via mail.

Continuous Delivery means the saving from the stored code to a repository. By doing that, the code can be used by other developers.

Continuous Deployment means the approval to release the data to the customer and to a productive system.

The entire process shall help reduce the time effort of the developers by replacing manual tests with automatic tests. Also a faster development and reaction on collisions improve the way. Contra is the high initial effort in creating the automatic tests and costs in such a CI/CD system.

## 1.2. CI-Lifecycle




Picture 2: The CI-Lifecycle

Development or changes in source code trigger the CI to build the files. Then they get tested, if such tests exist. Finally there is a report created automatically about analytical data, stats and failure or success of the tests.

### 1.3. Deployment-Pipeline

How much of the Pipeline is actually used? Mostly, companies start using the Continuous Integration and stay with that part. The Pipeline can be drawn to the customer. Refer to 1.1.

CI/CD shall contain the entire control about the way of the application, from developer to customer. So, the entire development shall be secured in some way.

A Pipeline shows in a virtual way, which steps have to be executed and when.

## 2. Hudson/Jenkins

### 2.1. Jenkins as tool for CI

Hudson was developed by Sun Microsystems, later Oracle. In the early 2000, there was the repository forked and formed the now called “Jenkins”. It is written in Java and very famous nowadays.

Both applications had the same target: continuous integration. As open souce project there are a lot of alternatives, but Jenkins managed to keep one of the most wanted. Jenkins is fully based on webapplication, running as a server service. Default running on port 8080 on localhost. The simple installation is maybe another reason for its popularity. It is running on Max, Linux and Windows, even on Docker. The jobs are not depending on the master, how is the server itself is also called. There are a lot of plugins that you can download and use, so Jenkins is really flexible and adaptive.

### 2.2. Installation

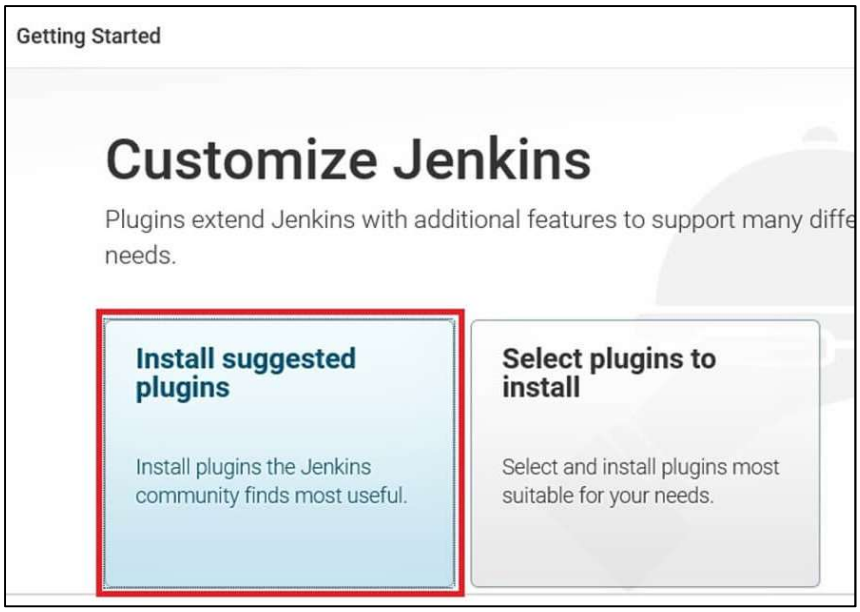
In this tutorial, Jenkins was installed on a Windows computer.

Download and install from here: https://www.jenkins.io/download/
Please check that the service “jenkins” is running.
The application is starting after installing or you manually enter in your browser:
http://localhost:8080/

Now you need to confirm the starting by entering the admin password.


Which can be found under the red marked path, meaning the path from your local storage system. Open the file and copy and paste the password into the field shown.

After installing the suggested plugins…




… you can “continue as admin”.

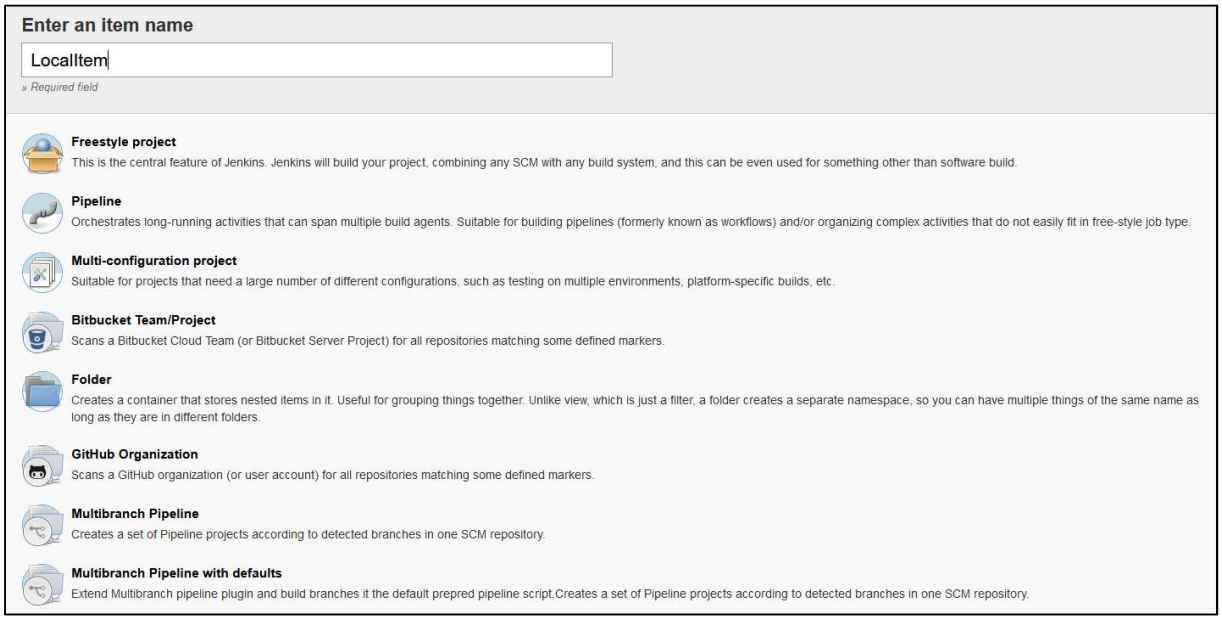

Therefore, please use the same password that you just entered in the unlock screen.
Username “admin”.

The initial view shall be looking like this:


And by seeing this view, the installation is done.

### 2.3. Create a job for local sources

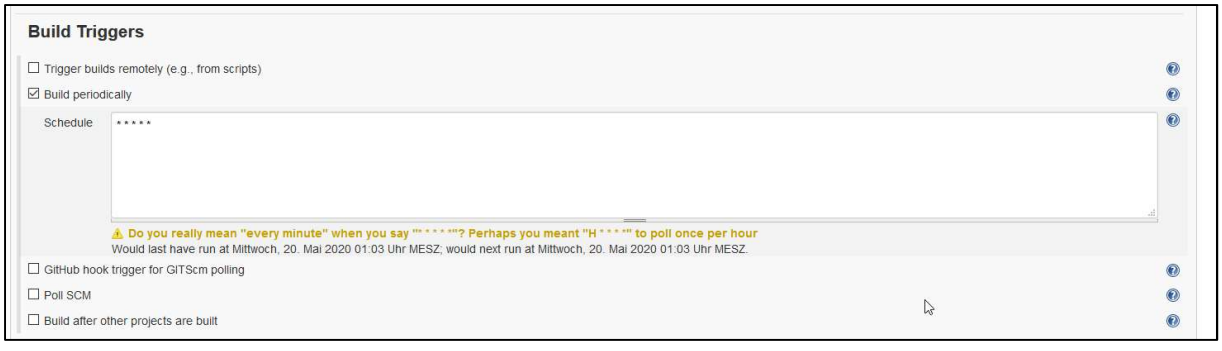
1. To create a job, you have to click on “New Item” on the left side.




Please give your job a name, e.g. “LocalItem”. Choose a freestyle project and continue with “OK”.

2. Fill with information

Alternatively give a description for your job. Tick in “Build Triggers” “Build periodically” and enter “* * * * *” to build your files every minute.



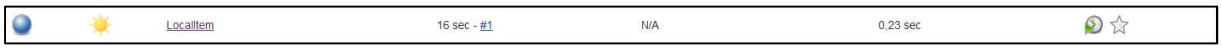
In “Build” click on “Add build step” and choose “Execute Shell”.


Enter the following:

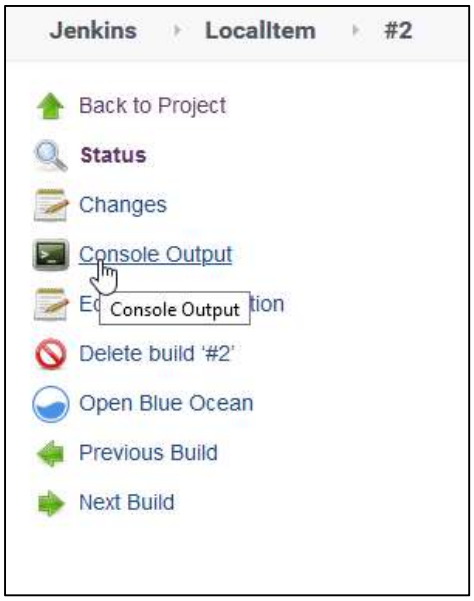
```
cd C:\Users\Labor\Documents\GitHub\e-portfolio-jenkins\myrogram\
javac myProgram.class
java myProgram
```

Click on “Save” to save the job. Now you can see the job in your workspace like this:


After some while you should see this:



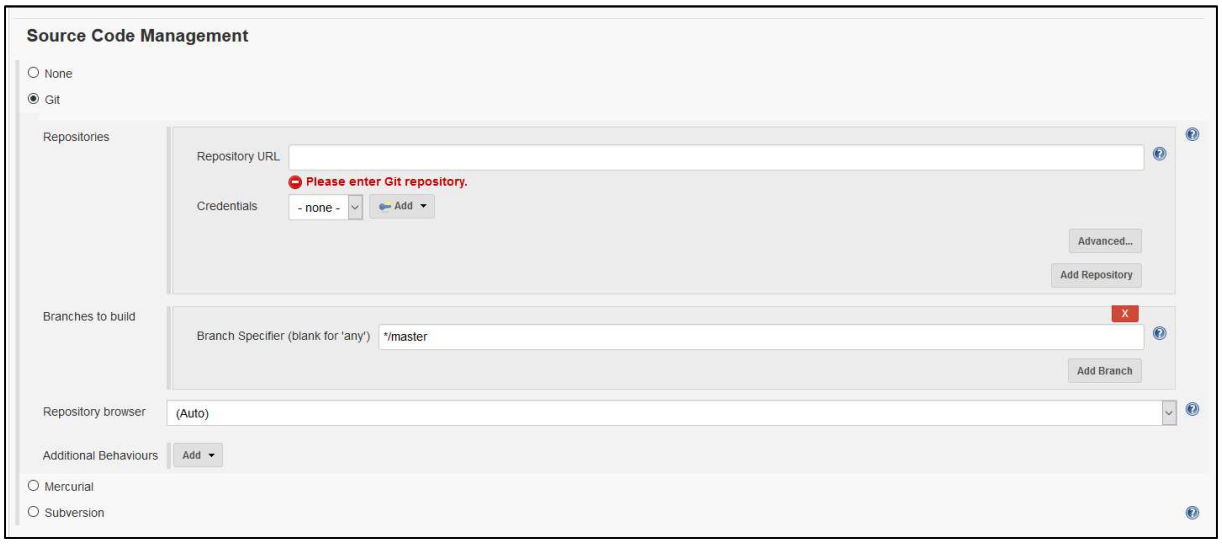

If the job is not blue, there are some errors in the test. Please open the job and have a look at the “Console Output” for your errors.

### 2.4. Create a job for remote sources on GitHub

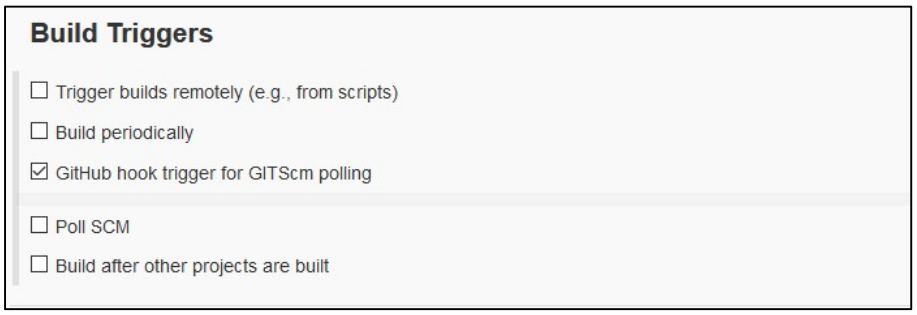
For this, please refer to step 1 of the previous chapter, then:

1. Fill with information




You need to enter your GitHub repository. Please copy and paste the link to your repository that you can find on GitHub:

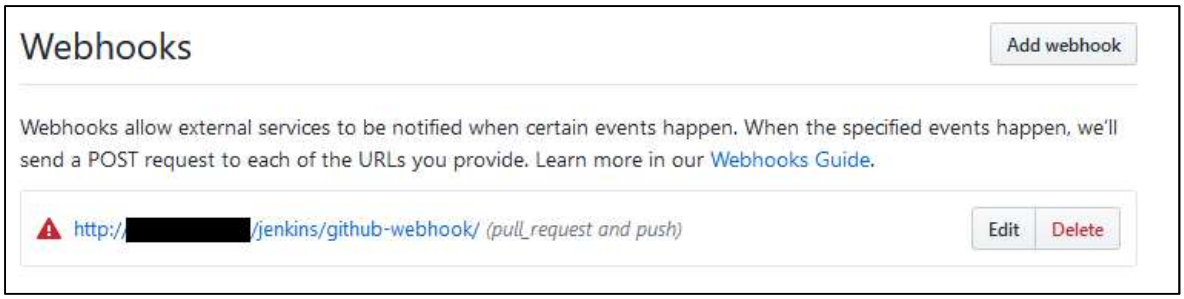

Change “Build Triggers” to GitHub.




And as described in the previous chapter, choose “Execute Shell” in “Build” and enter the following:

```
cd C:\Users\Labor\Documents\GitHub\e-portfolio-jenkins\myrogram\
javac myProgram.class
java myProgram
```

Click on “Save” to save your job. The job will now fail because we have to add the trigger for polling. Therefore open your GitHub-Repository, click on “Settings” and “Webhook” to “Add” a new Webhook.




In the black square enter your IPv4-address, pay attention to add the “/jenkins/github-webhook/”.

After “Build now” on Jenkins or after a change on the repository, you should see the following:


The job was executed with success.

### 2.5. Plugins and Tipps

Green Balls, different view for people with red green weakness.
Email-Plugin, advanced possibilities on writing mail notifications.
Timestamper, creates timestamps on the output of the job.
CI-Game Plugin, gives credit points for successful, takes credit points failed jobs.
Blue Ocean, makes Pipelines more easy to handle.

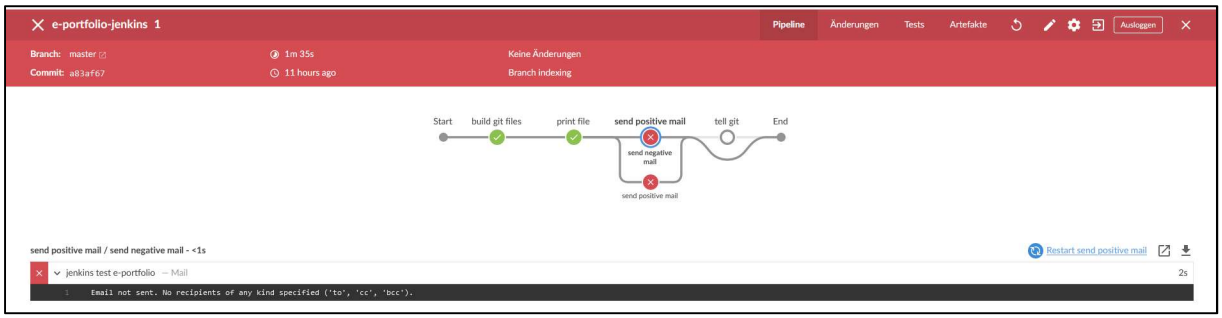
Jenkins can be connected with Docker, GitHub, Jira and more. The slaves should be easily changed. A slave contains the source code. A master is the Jenkins server. In a big company, you should definitely use multiple masters.

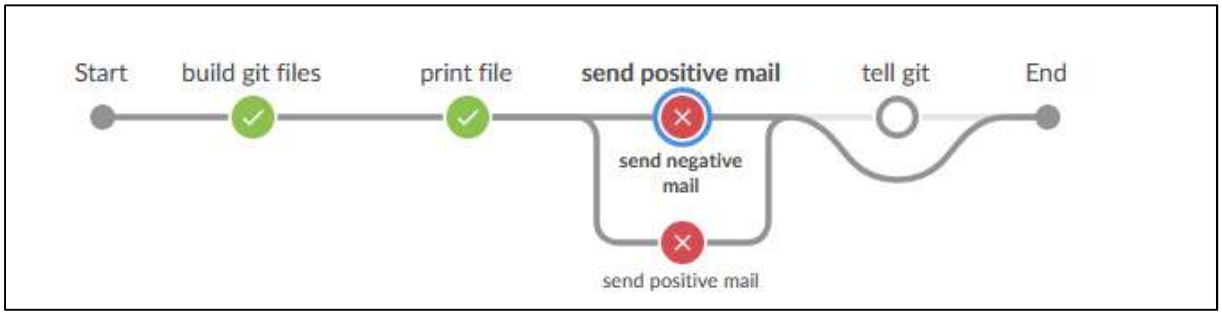
### 2.6. Create a Pipeline with Blue Ocean

Navigate over “Manage Jenkins” to “Manage Plugins” to the tab “Available” and search with the tool on the right “Blue Ocean”. Install the first entry.

After, you should see the following:

Click on the “Open Blue Ocean” link and create with “Neue Pipeline” a new Pipeline. Enter the properties in the guided mode and save your configuration. You can add steps via “Add Item”. Build the Pipeline. The following picture shows the success of the build and the printing of a text file. But it was not possible to send a mail, so “tell git” and “End” were never executed.




So, the project manager is informed on the status of his project. He can see that there was a problem with “send positive mail”.