

**GCP – HOL -Session 18**

# **Continuous integration with Jenkins**

## 1. Installation and setup of Jenkins

For most platforms you have native packages, see the [Jenkins Homepage](https://jenkins.io/).

### 1.1. Installing of the Jenkins server on Ubuntu

Jenkins provides Debian/Ubuntu packages which install Jenkins and register Jenkins as start service. See the [Install Jenkins on Ubuntu description](https://jenkins.io/doc/book/installing/#debian-ubuntu) The Linux installation creates a /etc/init.d/jenkins script which starts Jenkins automatically at boot time.

Jenkins stores all the settings, logs and build artifacts in its home directory. The default installation directory is /var/lib/jenkins under Ubuntu.

#### 1.1.1. Using the .war file to start Jenkins

Download the jenkins.war file from [Jenkins Homepage](https://jenkins.io/). From this file you can start Jenkins directly via the command line with java -jar jenkins\*.war.

If you start it locally, you find it running under the following URL: <http://localhost:8080/>

To run it in your Tomcat server, put the .war file into the webapps directory. If you start Tomcat, your Jenkins installation will be available under

http://localhost:8080/jenkins

|  |  |
| --- | --- |
|  | If the jenkins.war is deployed in your webapps directory, but  cannot be started and the tomcat manager says ﻿  FAIL - Application at context path /jenkins could not be started,  you may need to grant the permissons for ﻿JENKINS\_HOME.  ﻿sudo mkdir .jenkins  ﻿Sudo chown tomcat7:nogroup .jenkins  This makes the .jenkins folder writable and Jenkins can use it. |

## 2. Configure Jenkins

### 2.1. Connect to Jenkins for the initial configuration

After installation, open a browser and connect to it. The default port of Jenkins is :8080, therefore on your local machine you find it under the following URL:

http://localhost:8080/

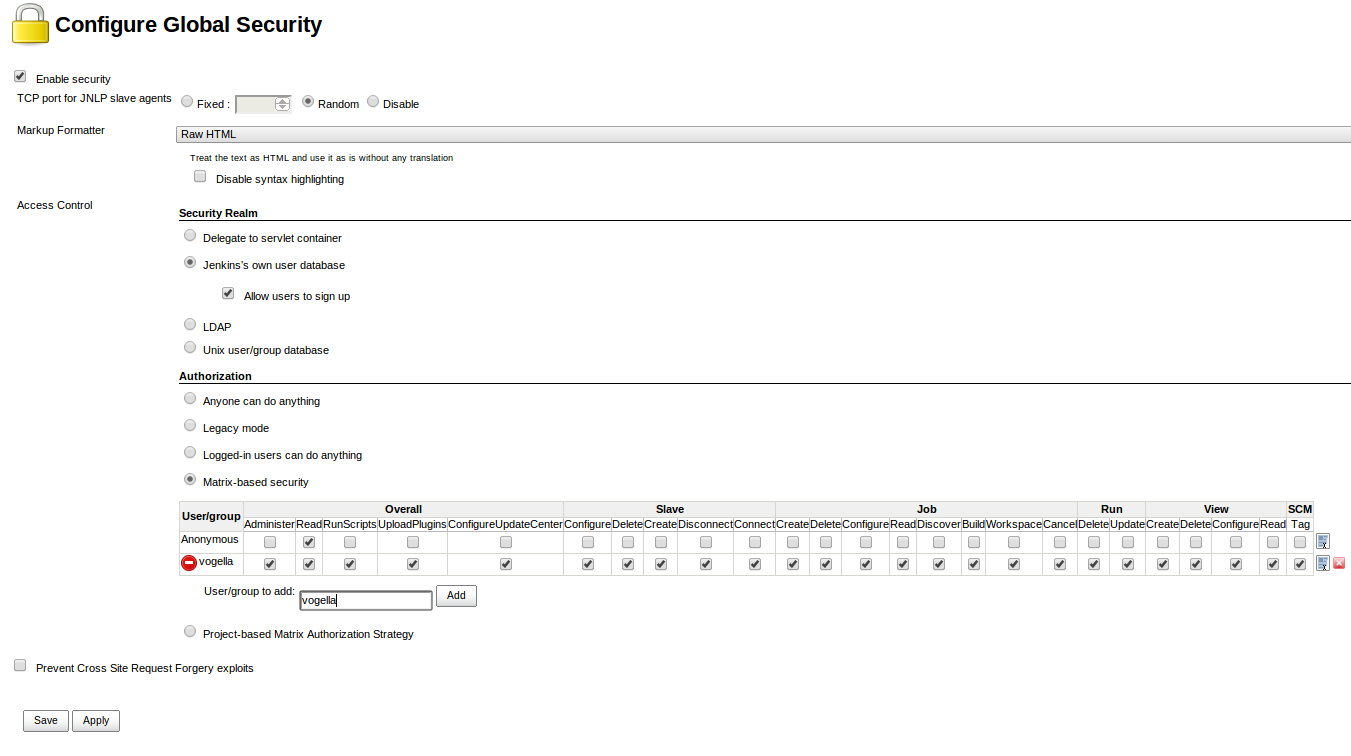
You will need to copy the initial password from the file system of the server.

Afterwards you can select to install plug-ins. Select the Install suggested plug-ins to get a typical configuration.

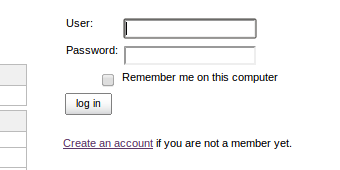
Create an admin user and press Save and Finish.

### 2.2. User management Jenkins

Select Manage Jenkins and then Configure Global Security. Select the Enable security flag. The easiest way is to use Jenkins own user database. Create at least the user "Anonymous" with read access. Also create entries for the users you want to add in the next step.



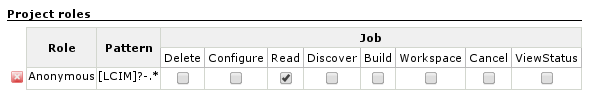
On the login page, select Create an account to create the users you just gave access.



### 2.3. Create a new user

Go to Manage Jenkins, Manage and Assign Roles and then Assign Roles to grant the newly created user additional access rights.

Navigate to Manage Roles to define access restrictions in detail. Pattern is a regex value of the job name. The following grants unregistered users read-only access to your build jobs that start with the C-MASTER or M-MASTER prefix and only those.



### 2.4. Generate ssh key for Jenkins user

If you want to access a private Git repo, for example at Github, you need to generate an ssh key-pair. Create a SSH key with the following command.

sudo -u jenkins ssh-keygen

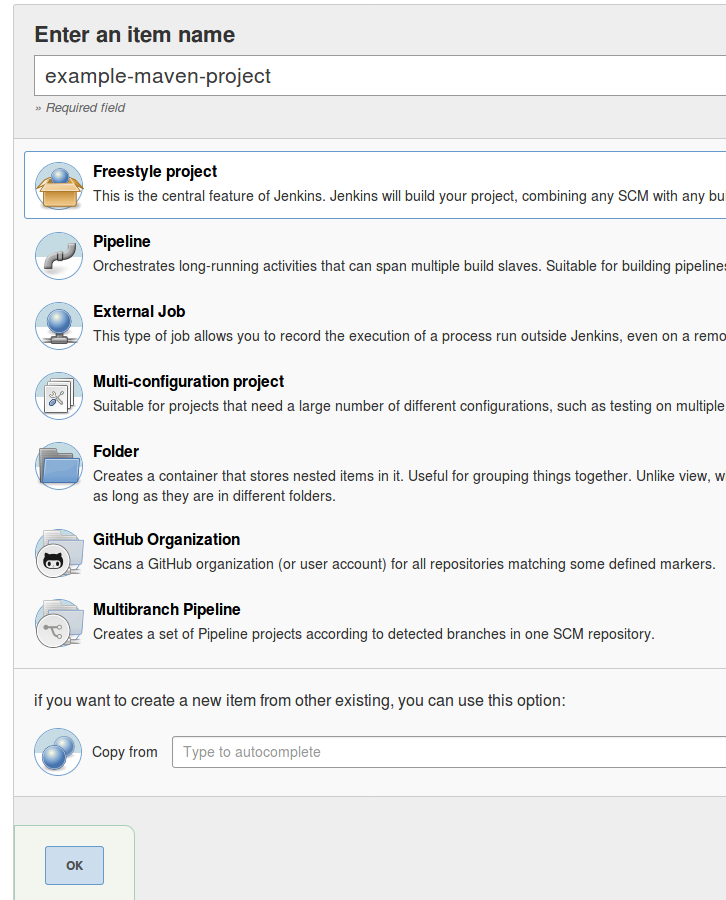
The public key must be uploaded to the service you are using, e.g., Github.

### 2.5. Configure the default port of the Jenkins build server

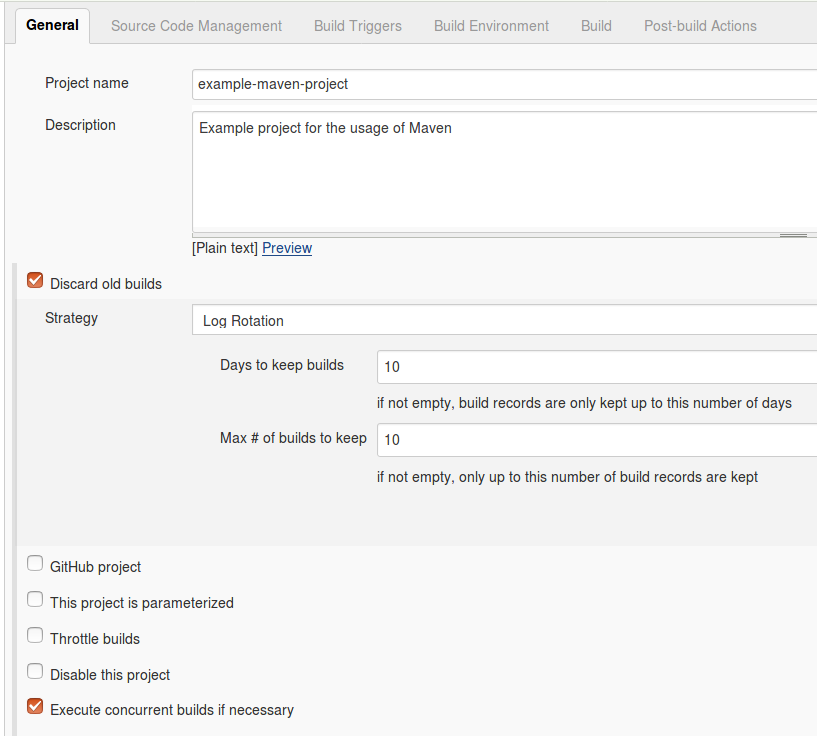
sudo vim /etc/default/jenkins

## 3. Setting up a Jenkins job

The build of a project is handled via jobs in Jenkins. Select **New Item**. Afterwards, enter a name for the job and select Freestyle Job and press OK.

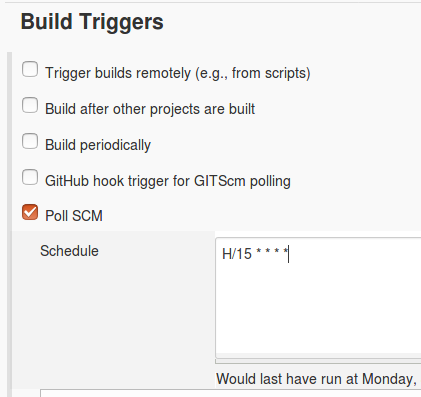


Enter a description for the job and configure how many old jobs should be retained.

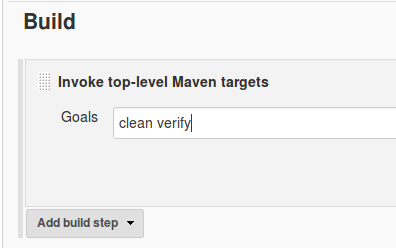


Configure how the source code can be retrieved. If you for example using Git, enter the URL to the Git repository. If the repository is not public, you may also need to configure the credentials.

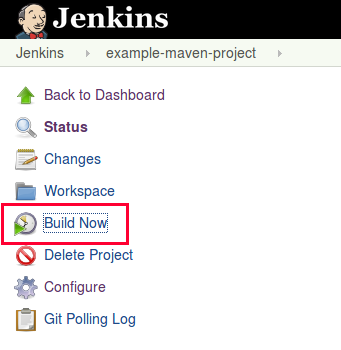
Specify when and how your build should be triggered. The following example polls the Git repository every 15 min. It triggers a build, if something has changed in the repo.



I typically delete the workspace before a build to avoid any side-effect. In the Build section you can add a build step, e.g., a Maven build.



Press Save to finish the job definition. Press Build Now on the job page to validate the job works as expected.



After a while the job should go to green or blue (depending on your configuration), if successful. Click on the job and afterwards on Console Output to see the log file. Here you can analyze the build errors.

## 4. Build Pipelines

Jenkins pipelines help you align the build process of a project. This is done by specifying tasks and the order in which they are executed. There are all kinds of possible tasks that a jenkins pipeline can do for you. For example, build assets, send an email on error or send the build artifacts via SSH to your application server.

### 4.1. Setup using a Jenkinsfile

Jenkins allows to specify pipelines using a Jenkinsfile. This is just a textfile that contains the necessary data for jenkins to execute the pipeline. It is called Jenkinsfile (notice: no file extension) and should be placed in the root of your project.

|  |  |
| --- | --- |
|  | This file should be checked into version control as it is needed on your Jenkins instance. |

Jenkins supports two different syntaxes.

1. Declarative (since Pipeline version 2.5)
2. Scripted

For this HOL session we will focus on the declarative approach.

The following example shows a pipeline with 2 stages:

pipeline **{**

agent any

stages **{**

stage**('Build Assets')** **{**

agent any

steps **{**

echo **'Building Assets'**

**}**

**}**

stage**('Test')** **{**

agent any

steps **{**

echo **'Testing stuff...'**

**}**

**}**

**}**

**}**

|  |  |
| --- | --- |
|  | The agent directive tells jenkins to allocate a workspace and an executor for the pipeline. Without it, the pipeline is not valid and therefore required. |

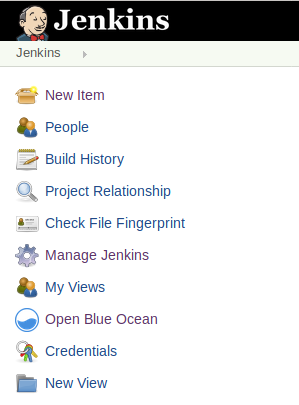
### 4.2. Setup using the Blue Ocean plugin

The above process can also be done using the [Blue Ocean](https://jenkins.io/projects/blueocean/) Jenkins plugin.

#### 4.2.1. Installation

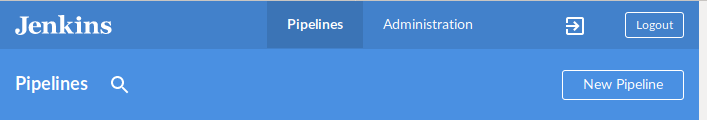
To install the plugin go to **Manage Jenkins**  **Manage Plugins**  **Available** and select the Blue Oceanplugin.

After the installation is finished you have an additional menu entry called Open Blue Ocean in your main Jenkins navigation.

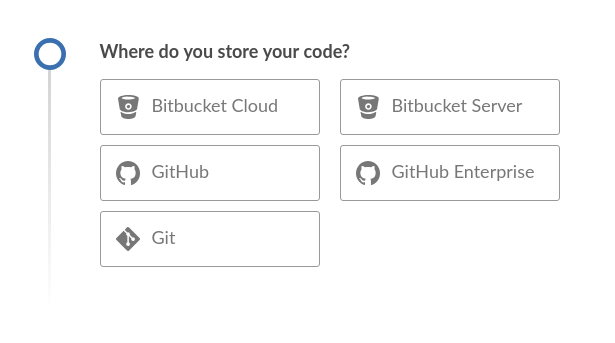


#### [5.2.2. Creating a new pipeline](https://www.vogella.com/tutorials/Jenkins/article.html#creating-a-new-pipeline)

Click on New Pipeline to create a new pipeline.



Select your version control provider.



|  |  |
| --- | --- |
|  | For this example we will use a github repository |

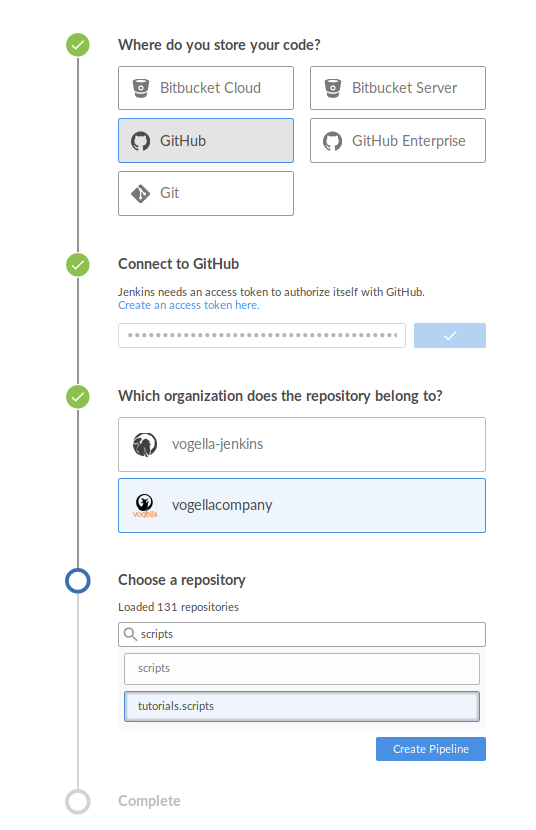
Depending on your provider you will need to pass some kind of credentials.

|  |  |
| --- | --- |
|  | Github provides the ability to generate access-tokens that applications can use to access the platform with your user. You can also restrict what the acess-token can do. |

The Blue Ocean application will provide a link to the GitHub page you need visit. The necessary permissions that Blue Ocean needs to operate are already selected. Add a description and click on Generate Token at the bottom of the page.

Copy the generated token and paste it in the Blue Ocean mask.

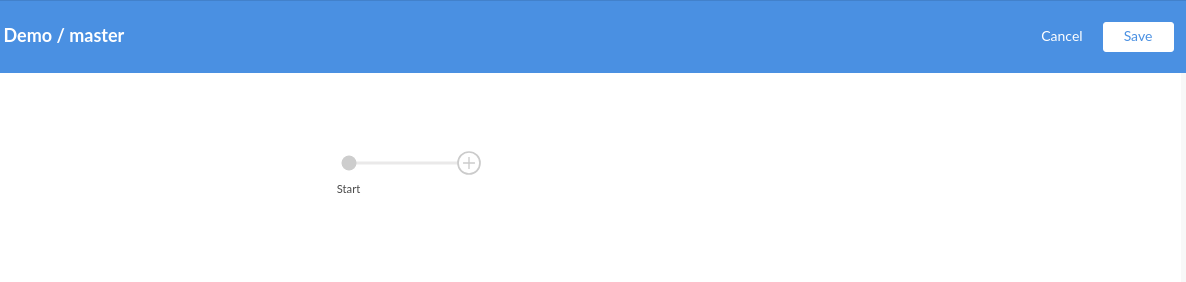
Select the account the repository belongs to and select the repository.



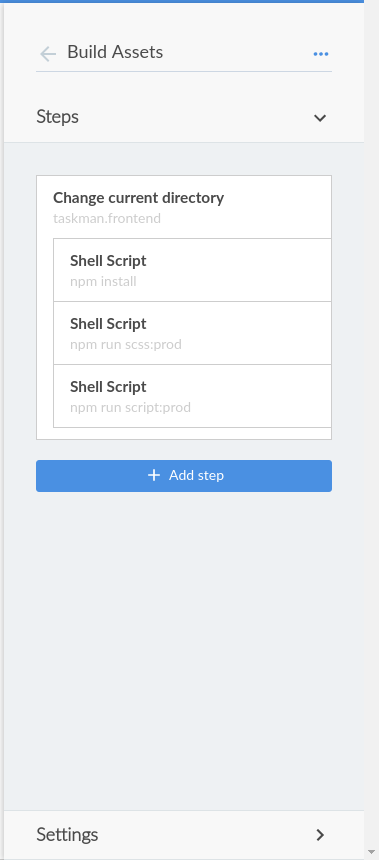
If you already have a Jenkinsfile with pipelines in the repository it will show up in the last step. If not Blue Ocean offers to create one for you.

#### 4.2.3. Adding steps to your pipeline

In the next screen you will see a visual representation of your pipeline. Here you can add or remove steps.



To create a new stage click on + in the canvas. A menu will open on the right that lets you specify a name and what steps you want to perform.



After you have finished editing the pipeline Blue Ocean offers to commit the newly created pipeline to your repository.

Under the hood Blue Ocean only created a valid Jenkinsfile for jenkins to use.

After committing Jenkins will build the project using the newly modified pipelines.