Setting up a Jenkins-Server and creating a Jenkins-Maven-Pipeline for Tomcat 6.0.53

Installing Jenkins

The following instructions are based on this site: https://pkg.jenkins.io/debian-stable/

1. To use the Debian package repository of Jenkins add it's key to the system by executing the following command via the terminal: wget -q -O https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add -
2. Add the following line to the file "sources.list" which can be found in /etc/apt/: deb https://pkg.jenkins.io/debian-stable binary/
3. Run the command "sudo apt-get update" and then the command "sudo apt-get install jenkins" (in that order)

You should be able to access Jenkins in a web browser now.

Getting started

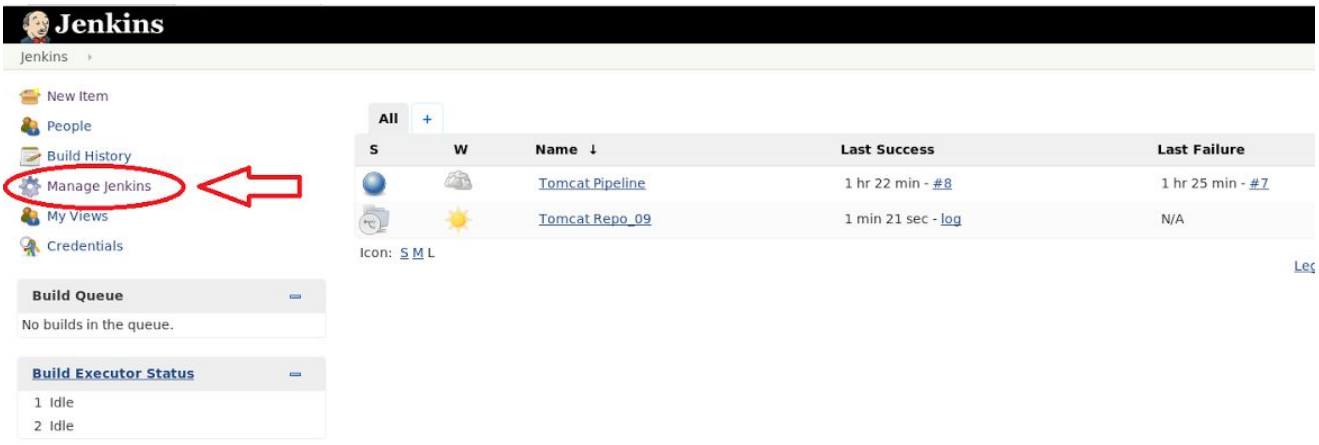
1. Open a web browser and type in localhost:8080
2. On your system navigate to /var/log/jenkins/ and copy the initial password from jenkins.log
3. Paste the password into the field in the web browser
4. Select plugins to install. Make sure the plugins "Pipeline" and "GitHub" are selected (most of the plugins should already be selected by default)

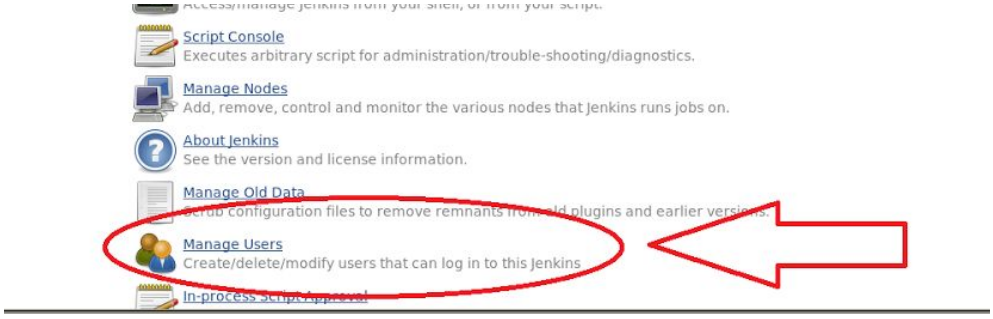
Creating A User

When you first login to Jenkins you use the initial password found in jenkins.log. To avoid having to type in the same long password again the next time you login, you should create another user.

In Jenkins:

1. Go to “Manage Jenkins” > “Manage Users”



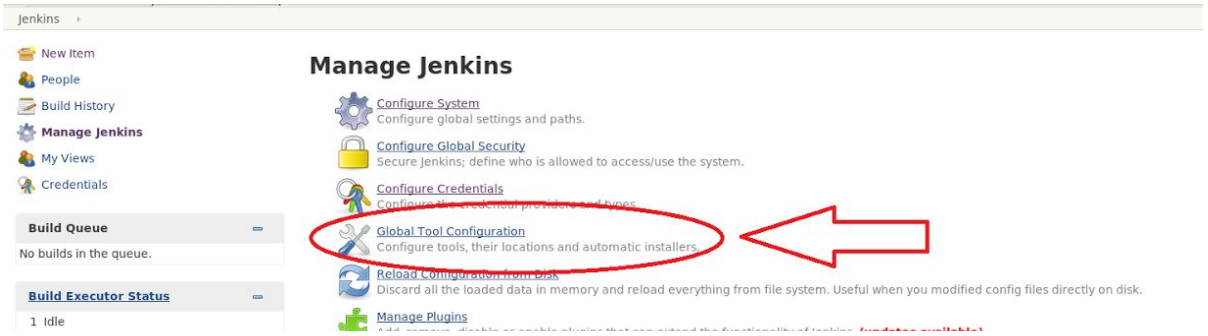


2. Fill in the input fields
3. Click “Create User”

Configure Maven-Tool and JDK

Maven does not come with Jenkins by default. Note that the following steps are not the only solution.

In Jenkins:
Got to “Manage Jenkins” > “Global Tool Configuration”



1. Maven
    a. Scroll down to section “Maven”
    b. Click on “Add Maven”
    c. Type in name for Maven tool and select “Install automatically” and version.
    d. Click “Apply” and “Save”
2. JDK
    a. Scroll down to section “JDK”
    b. Click on “Add JDK”
    c. Type in name for JDK and select “Install automatically” and version.
    d. Click “Apply” and “Save”

Creating Jenkinsfile for Tomcat 6.0.53

This is the heart of this exercise. The Jenkinsfile is one way of defining a Jenkins Pipeline. Here you define different stages through which your project will be built, tested and deployed. The Jenkinsfile has to be located in your project’s repository (the root directory should do).
In GitHub-Repository:

1. Create Jenkinsfile
2. Use the following structure:

```
pipeline {
    agent any

    triggers {
        pollSCM('*/5 * * * *')
    }

    tools{
        maven 'apache-maven-3.5.2'
        jdk 'java-sdk-1.8'
    }
    stages {
        stage('Build') {
            steps {
                sh 'cd Tomcat && mvn clean compile'
            }
        }
        stage('Test'){
            steps {
                sh 'cd Tomcat && mvn test'
            }
        }
        stage('Deploy') {
            steps {
                sh 'cd Tomcat && mvn assembly:single'
            }
            post {
                success {
                    archiveArtifacts(artifacts: 'Tomcat/target/*.jar', allowEmptyArchive: true)
                }
            }
        }
    }
}
```

“pipeline” - All valid Declarative Pipelines must be enclosed within a “pipeline” block.

“agent” - The agent section specifies where the entire Pipeline, or a specific stage will execute in the Jenkins environment depending on where the agent section is placed.

“Triggers” - Here one can specify a trigger for polling. The value for pollSCM stands for a poll for changes every 5 minutes.

“tools” - This is a section for the defining of tools to auto-install and put on the PATH. Note that these tools must be pre-configured manually in “Manage Jenkins” -> “Global Tool Configuration”

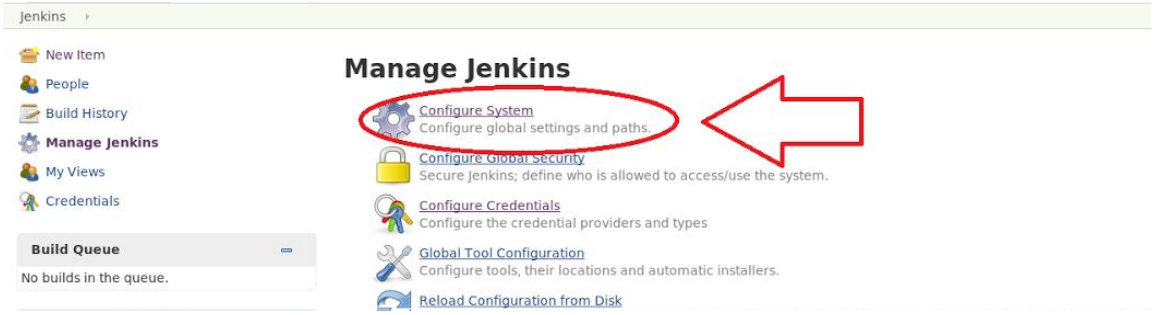
“stages”/”stage” - All the relevant work done by a Pipeline will be wrapped in one or more stage directives

“steps” - Steps are wrapped in stages and they take a block of Scripted Pipeline.

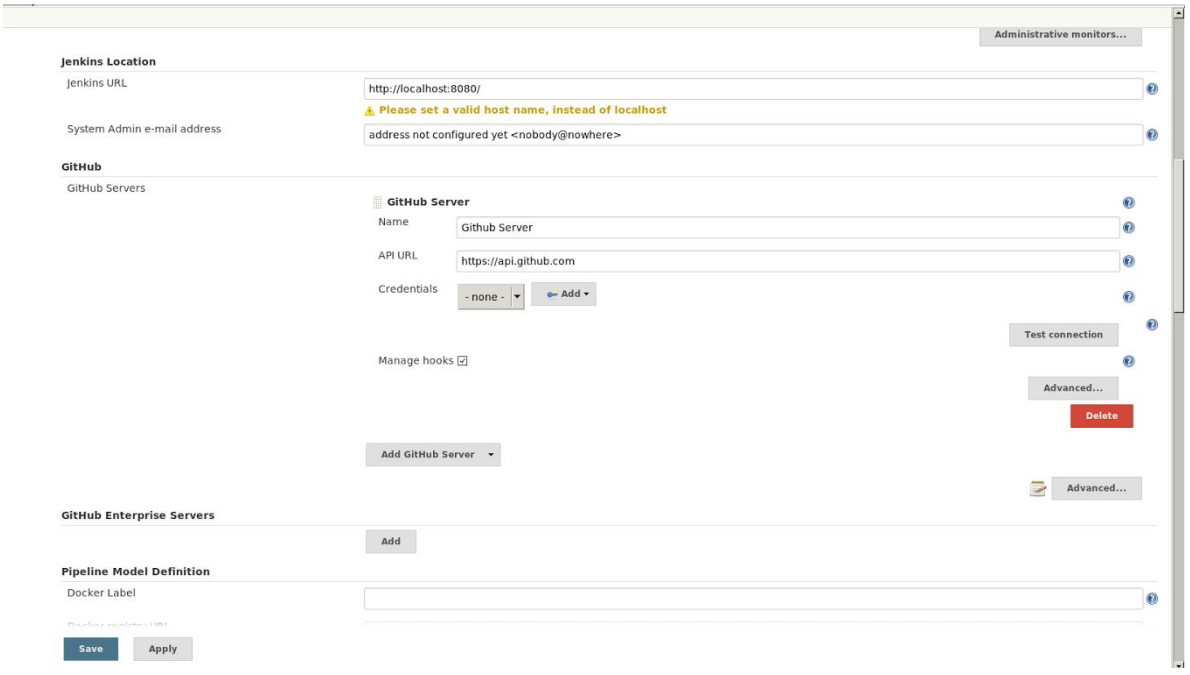
“post” - Defines action which will be run at the end of the Pipeline run or stage

Configuring GitHub for Jenkins

Got to “Manage Jenkins” > “Configure System”



1. Scroll down to section “GitHub” and click “Add GitHub Server”
2. Enter a Name and add credentials
    a. Make sure to set “Kind“ to ”Secret” Text. Here you should paste a token which you can generate here: https://github.com/settings/tokens into the field “Secret”

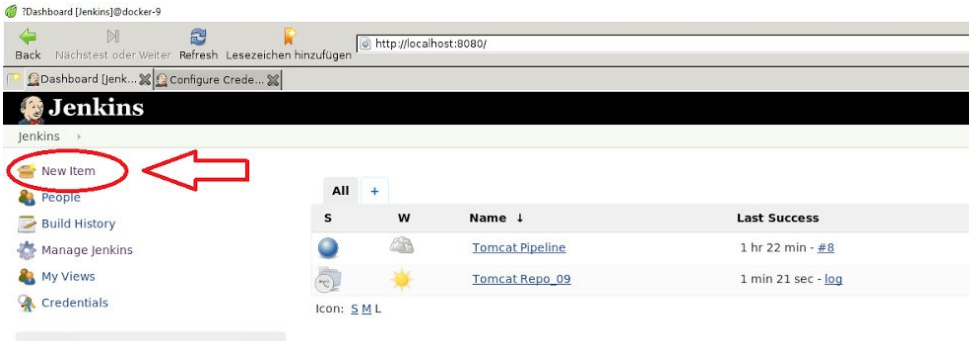


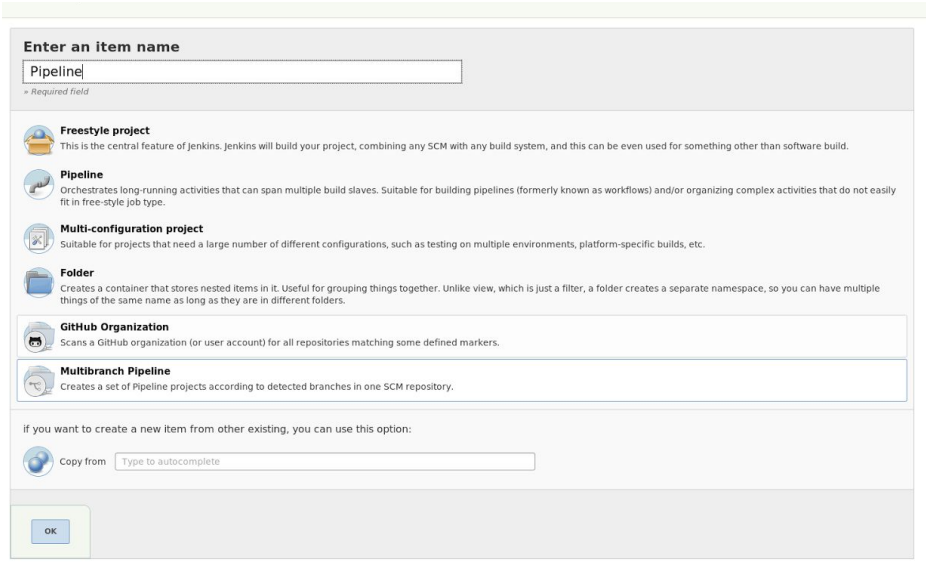
3. Click “Apply” and “Save”

Creating a Jenkins Pipeline for Tomcat 6.0.53 via Maven

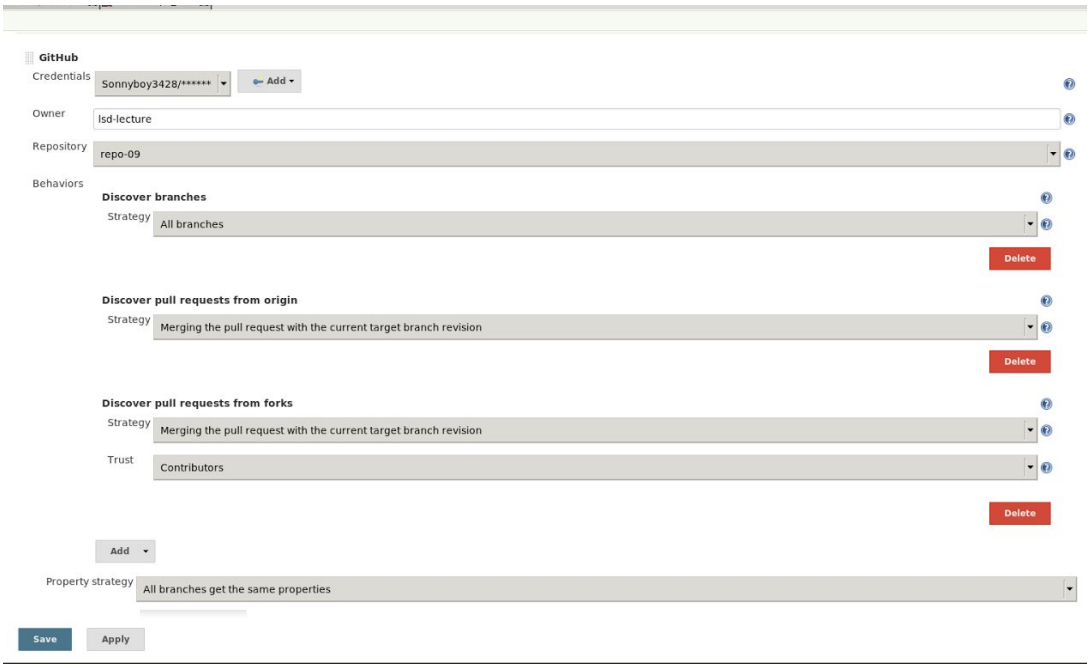
In Jenkins:

1. Go to “New Item”





2. Click on “Multibranch Pipeline”, enter a pipeline name and click on “OK”
3. Under section “Branch Sources” click “Add” Source -> “GitHub”
4. Add Credentials with “Kind” set to “Username and Password”
5. Type in name of the repository owner and chose the right repository



6. Click “Apply” and “Save”

Building a Jenkins Job

If the Jenkins Job was configured correctly you should witness the first build. You can monitor the success of your builds on the dashboard (starting page) or by clicking on the builds you would like to have a closer look at.