Installation Web Development Server Guide

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Name** | **Date** |
| Author |  | Duc NGUYEN | 28-Feb-2014 |
| Review |  |  | 03-Mar-2014 |
| Approval |  |  |  |

**Content**

[1. Introduction 4](#_Toc387822680)

[1.1 Target audience 4](#_Toc387822681)

[1.2 Glossary 4](#_Toc387822682)

[1.3 References 4](#_Toc387822683)

[2. Install NodeJS 5](#_Toc387822684)

[3. Install Ruby AND Compass 6](#_Toc387822685)

[4. Install Git 7](#_Toc387822686)

[5. Install Chrome 8](#_Toc387822687)

[6. Install and Configure Hudson 10](#_Toc387822688)

[7. Remaining problems 20](#_Toc387822689)

[7.1 Bower not run correctly on Server 20](#_Toc387822690)

**Changes**

|  |  |  |
| --- | --- | --- |
| 1.0 | 21-Feb-2014 | Initial version |
|  |  |  |
|  |  |  |
|  |  |  |

# Introduction

This is document for installation web development server.

Server descriptions: Fedora release 18 (Spherical Cow).

There is several ways to install Hudson. In this document we install Hudson on Jboss Application Server 7.1

## Target audience

This document is intended primarily for integrators and application developers.

It does not address non-technical people and customers.

Its distribution must remain internal and confidential.

## Glossary

|  |  |
| --- | --- |
| User credentials | Establish the identity of a user, for example in the form of username and password. |
| OS | Operating System |
| npm | Node package management |

## References

**InstallationWebDevelopmentServerGuide-***<version>***.docx**

# Install NodeJS

Node.js and npm are available in Fedora 18 and later. Just use your favorite graphical package manager or run this on a terminal to install both npm and node:

**$sudo yum install nodejs npm**

Test installation successful:

**$node --version**

**$npm --version**

# Install Ruby AND Compass

* Install Ruby

CentOS, Fedora, and RHEL use the yum package manager. You can use it like this:

**$ sudo yum install ruby**

Check installation:

**$ruby --version**

* Install Compass

**$gem update --system**

**$gem install compass**

Check installation:

**$gem --version**

# Install Git

If you want to install Git on Linux via a binary installer, you can generally do so through the basic package-management tool that comes with your distribution. If you’re on Fedora, you can use yum:

**$ yum install git-core**

Test:

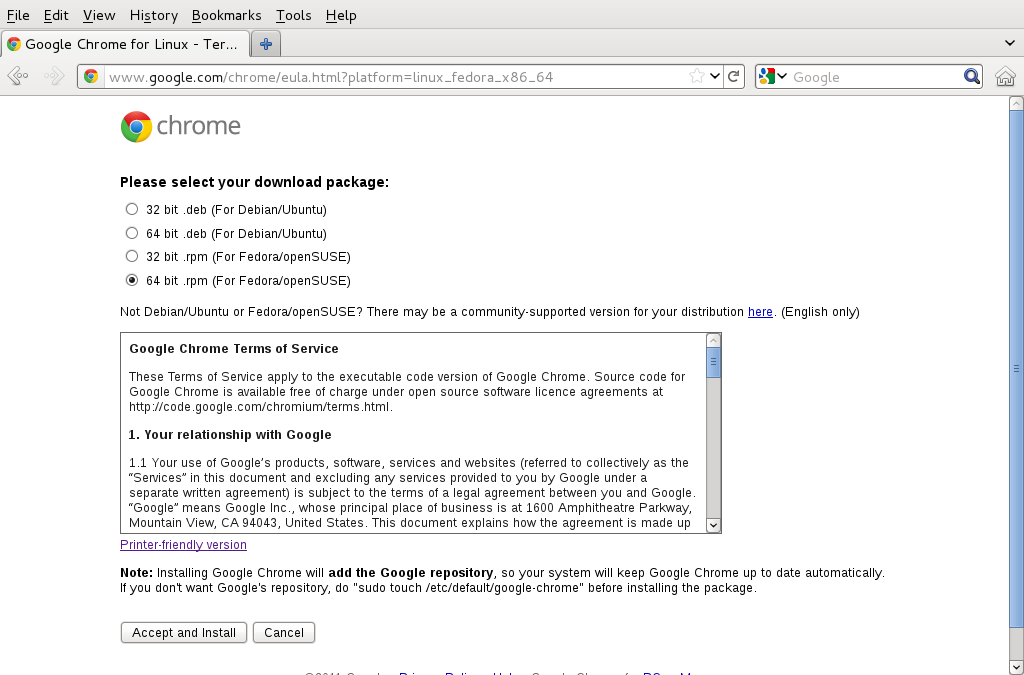
$**git --version**

# Install Chrome

Here we install Google Chrome because our project testing on Google Chrome. If you using other browser for testing like Firefox or Opera, you don’t need to read this part.

First go to the link: <https://www.google.com/intl/en/chrome/browser/>

Get the installer package suite with you OS version.



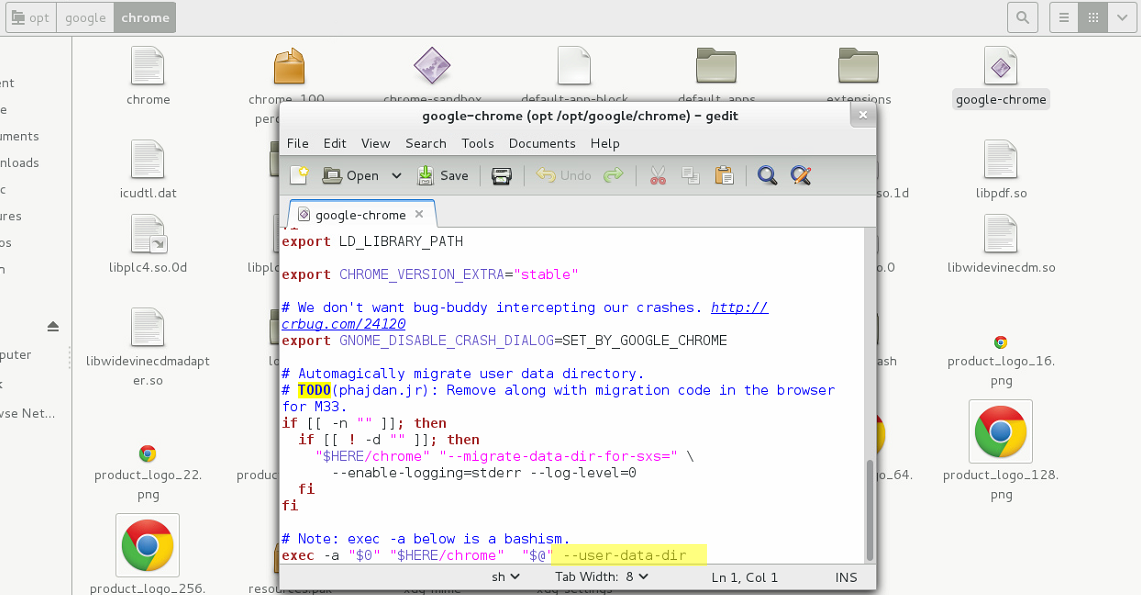
After installed. If you using root user for running server. Maybe you can’t run Google Chrome. Because Google Chrome doesn’t run with root user by default.

Try to do some work-around below:

In Fedora we have Google Chrome installed in: /opt/google/chrome

Open file google-chrome with any kind of text editor you like most.

Add the line “--user-data-dir” like the picture below.

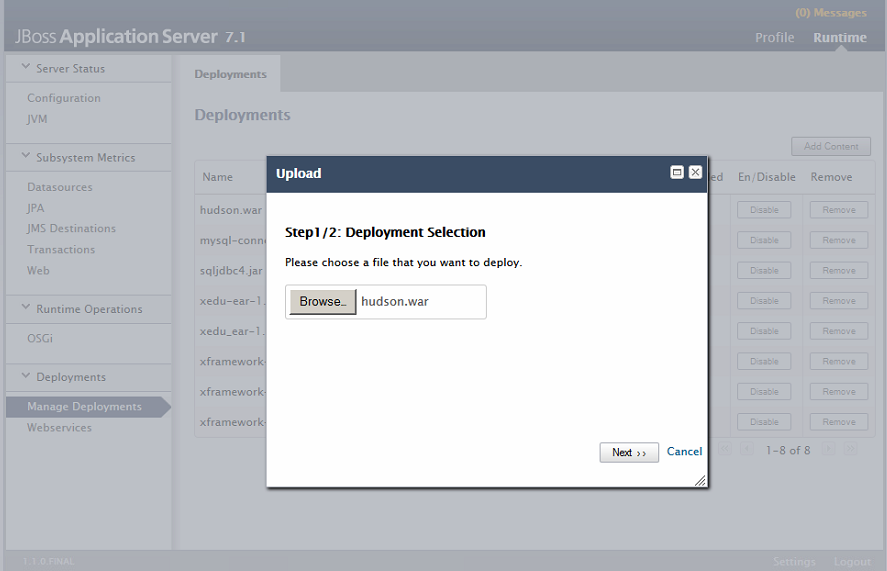


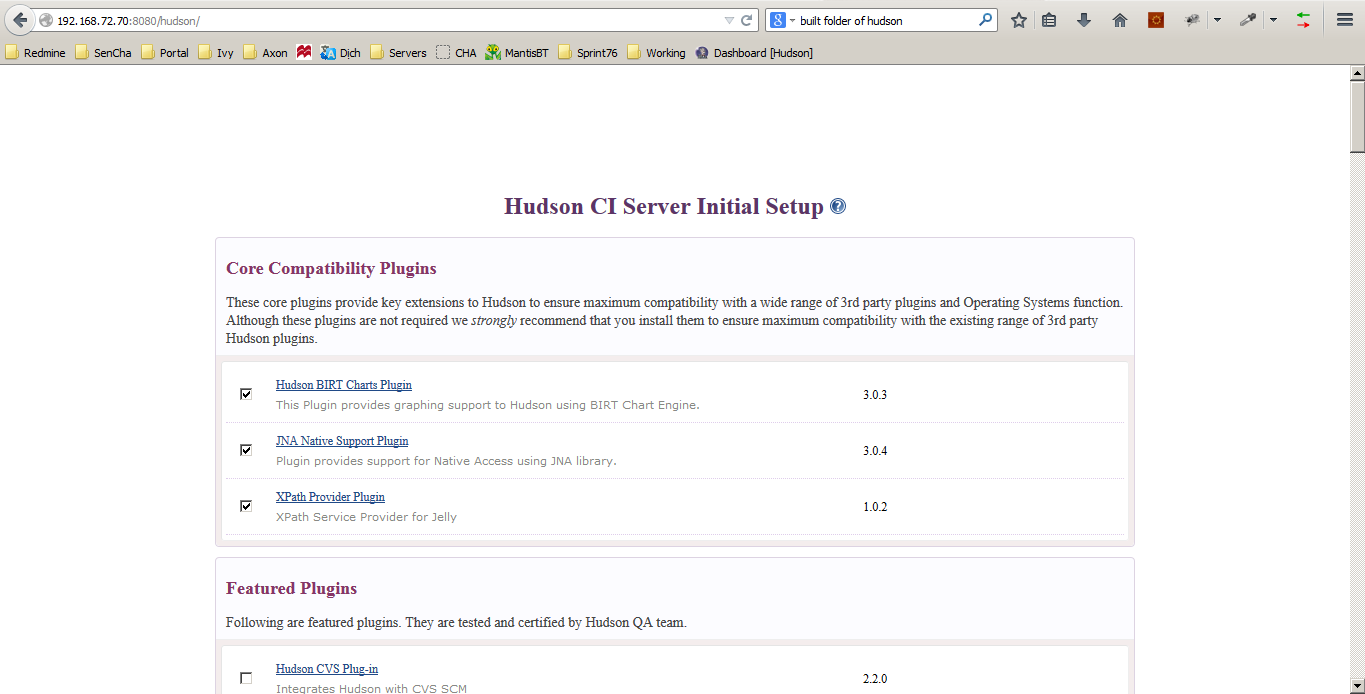
Test installation:

Open the Google Chrome and hit: <http://www.ti-informatique.com/>

# Install and Configure Hudson

There is several ways for install Hudson. In this document we just install Hudson by deploy the Hudson.WAR file to Jboss Application Server 7.1

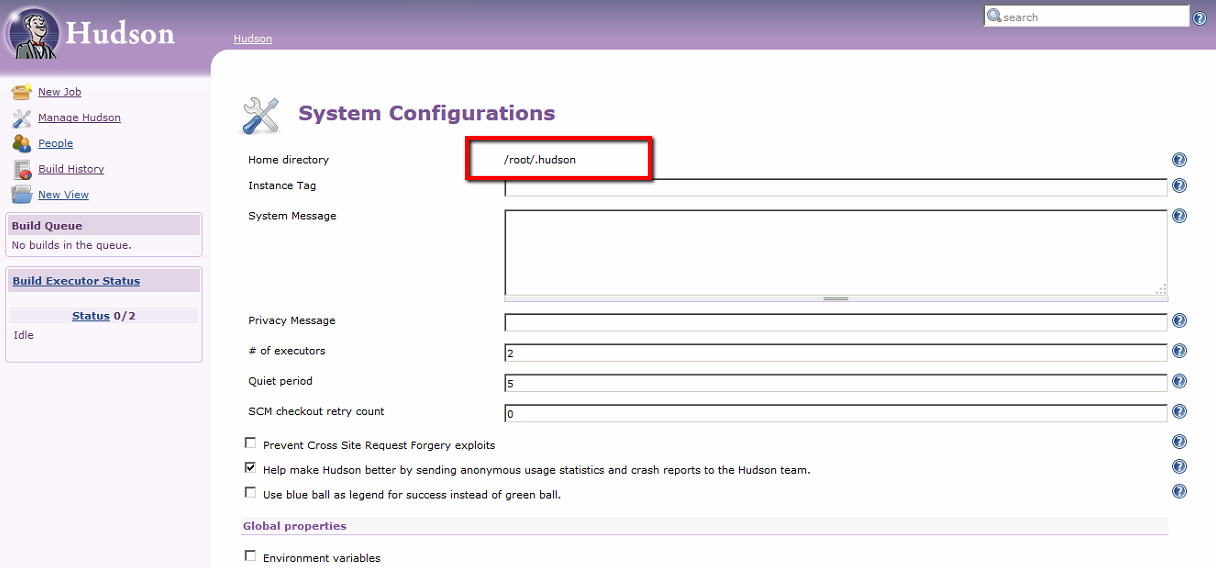
1. Download the Hudson war file from homepage of Hudson: <http://hudson-ci.org/>.
2. Deploy on Jboss:  
   
3. Open the Setup screen of Hudson:



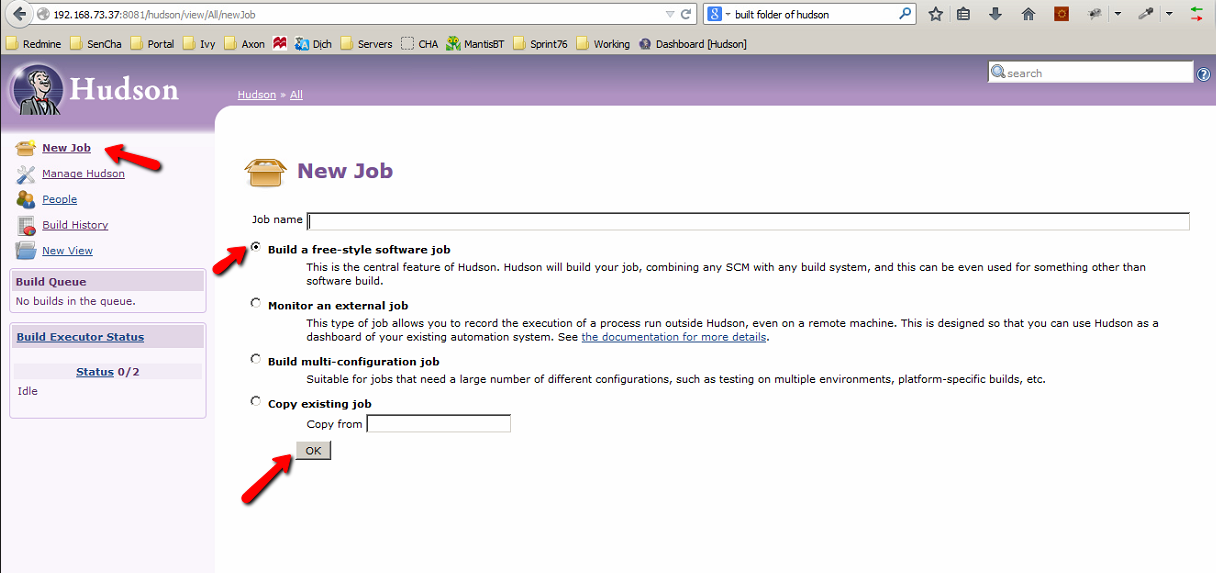
For me I don’t know much about Hudson, and I check all options in **Core Compatibility Plugins** and **Featured Plugins.**

1. After installing finish:

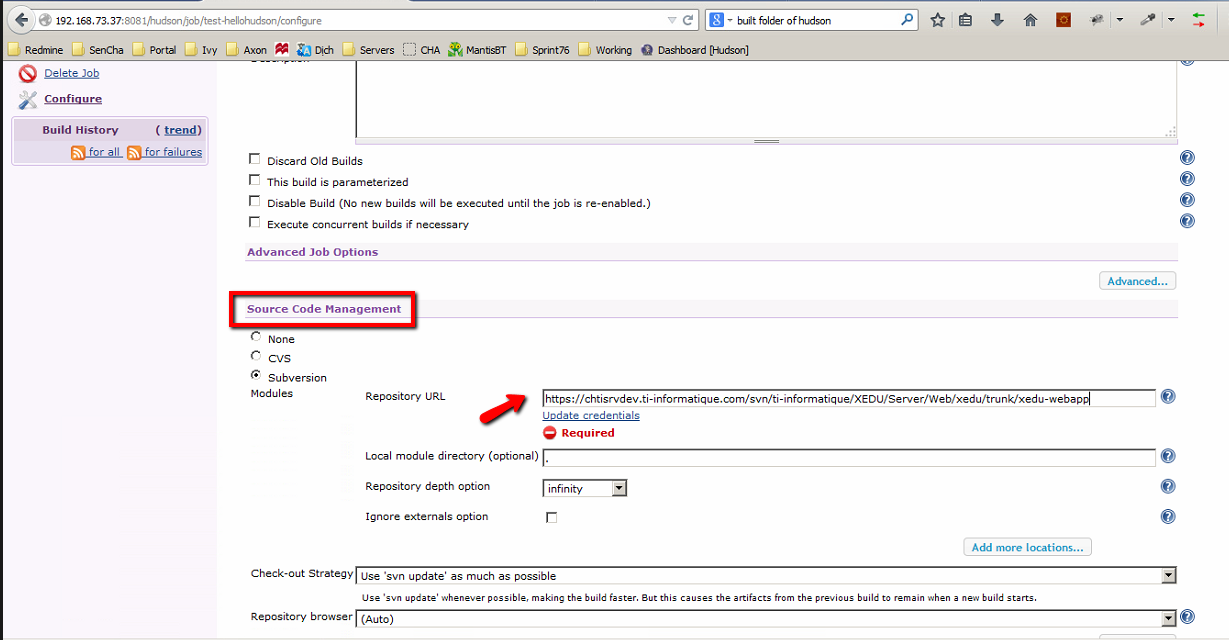
Click Manage Hudson 🡪 System Configuration:



Remember the Home directory. All our jobs will store in this directory.

1. Now all about Hudson setting are finish. Let’s start with xedu-webapp project.  
   

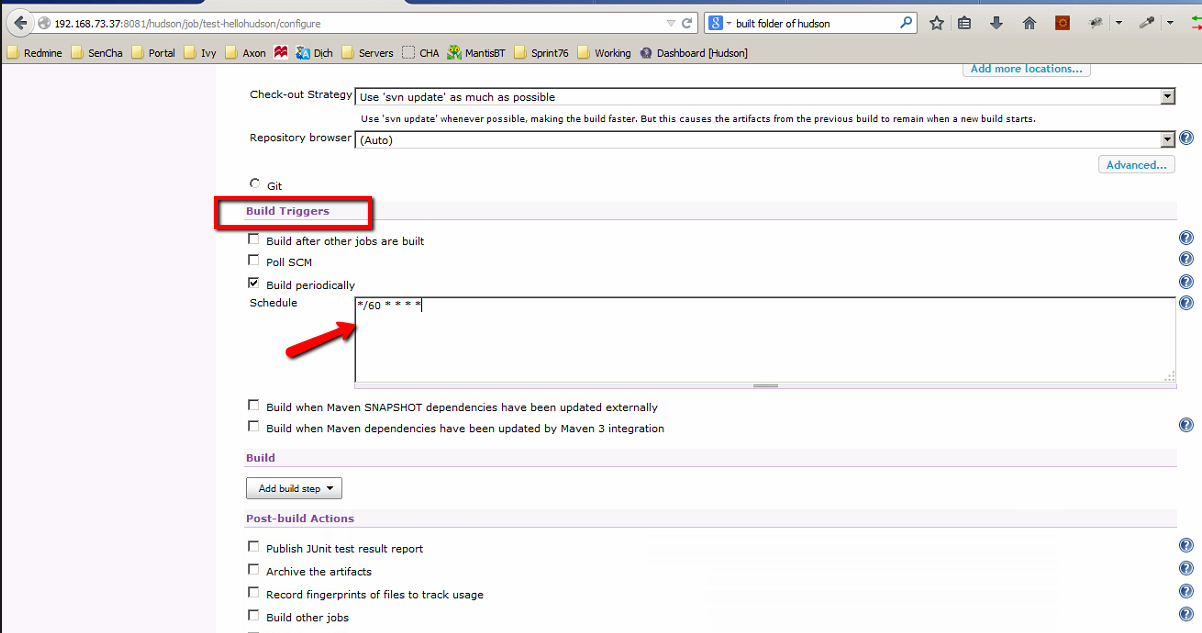
Input the job name and click OK.

1. Check out code from Source Code Management:  
   

This step may require user credentials for access SVN server to get source code.

Input your credentials needed.

1. Scheduling the job:



In this example: **\*/60 \* \* \* \*** mean: build every 60 minutes of everyday.

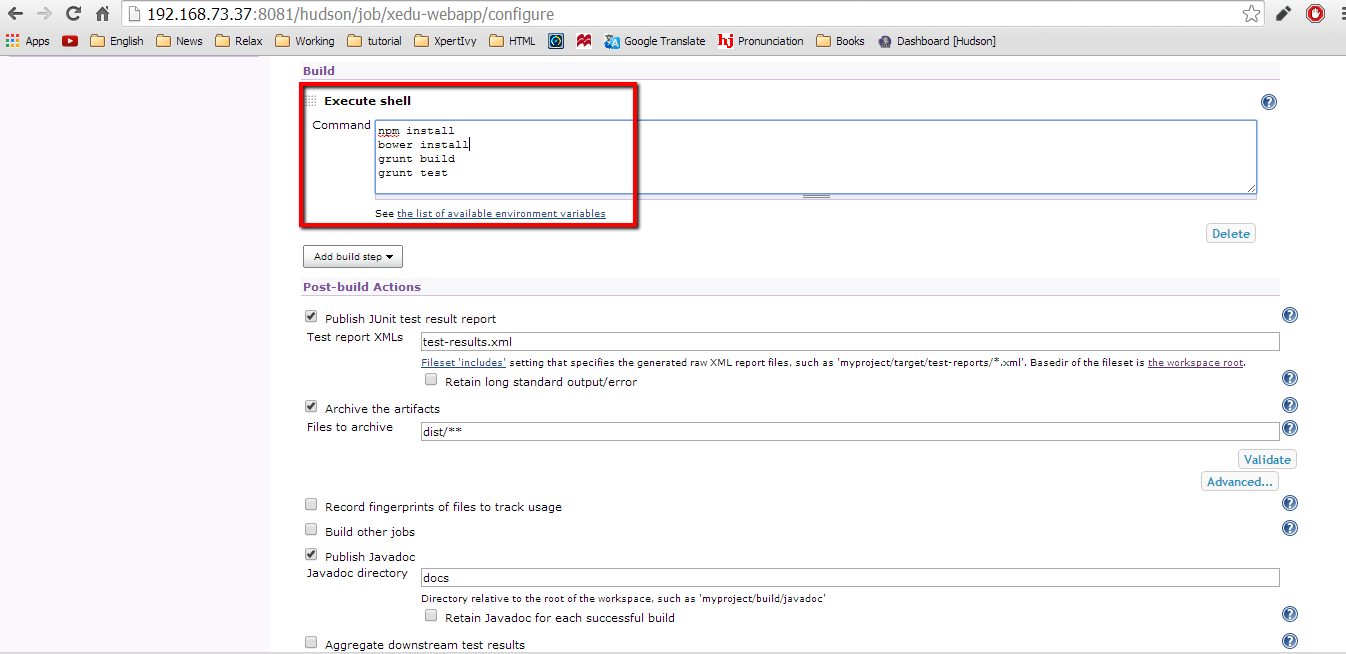
1. Configuration Build.  
   In build part: Click **Add build step** 🡪 Choose **Execute shell**

**npm install**

**bower install**

**grunt build**

**grunt test**



1. Click **save** at the bottom of page and Click **Build Now**.

If your build successful you will have a screen like below.

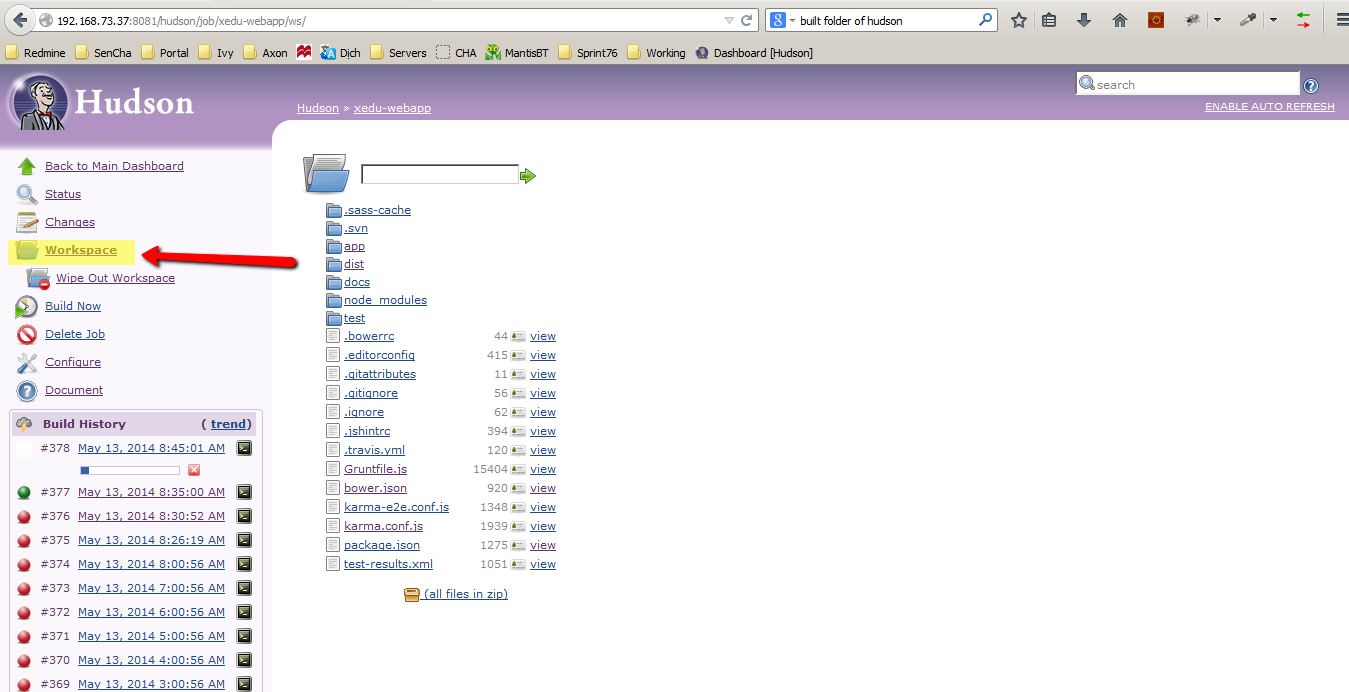


Figure 1.Build successful, we have dist folder

If your build FAILURE this may causes by:

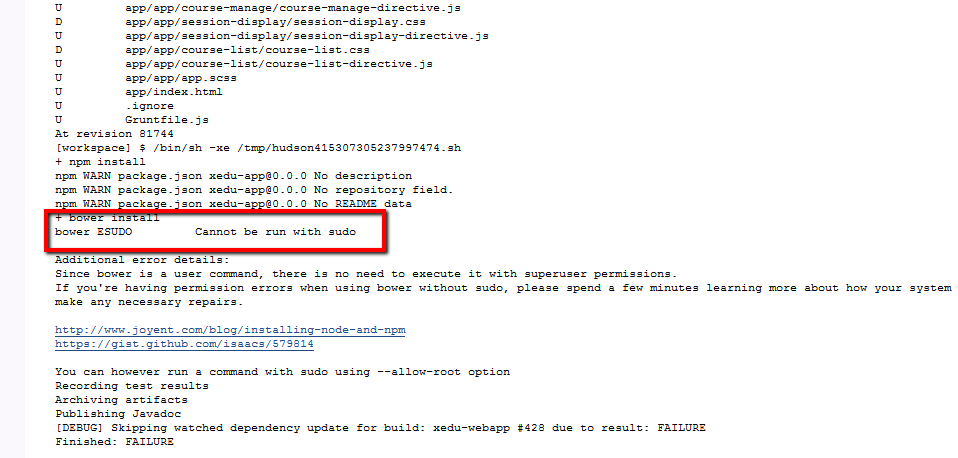


Figure 2.Bower not run with root user

Replace the command **bower install** by **bower install --allow-root**

Otherwise you may have the task **bower install** running for too long, read **remaining problems** partto solve it.

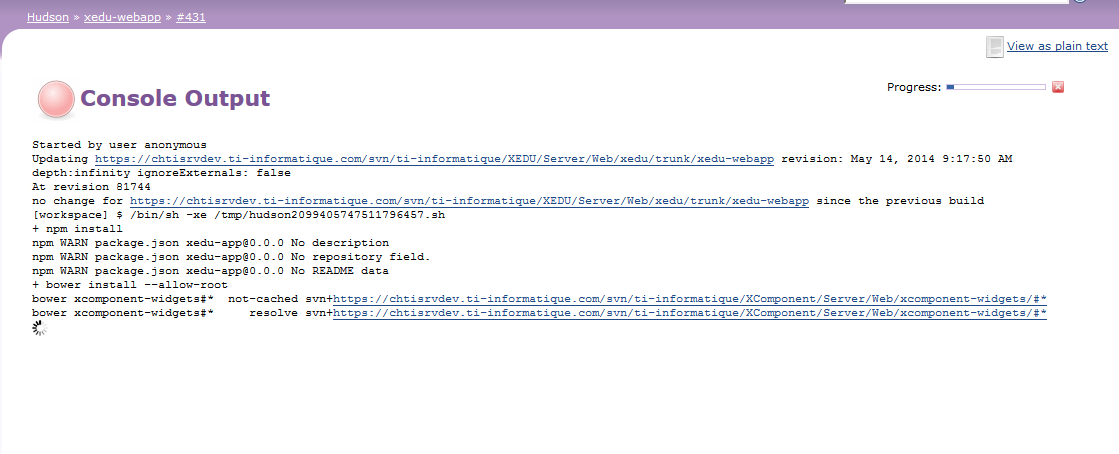
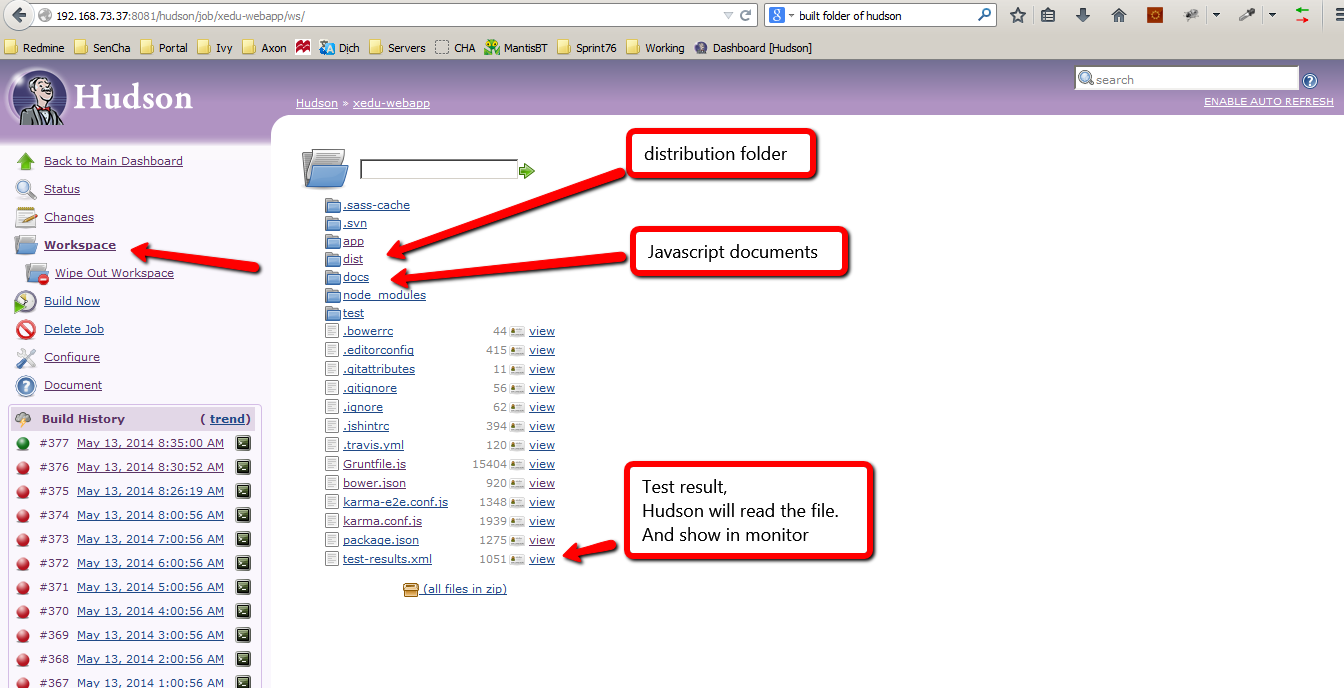


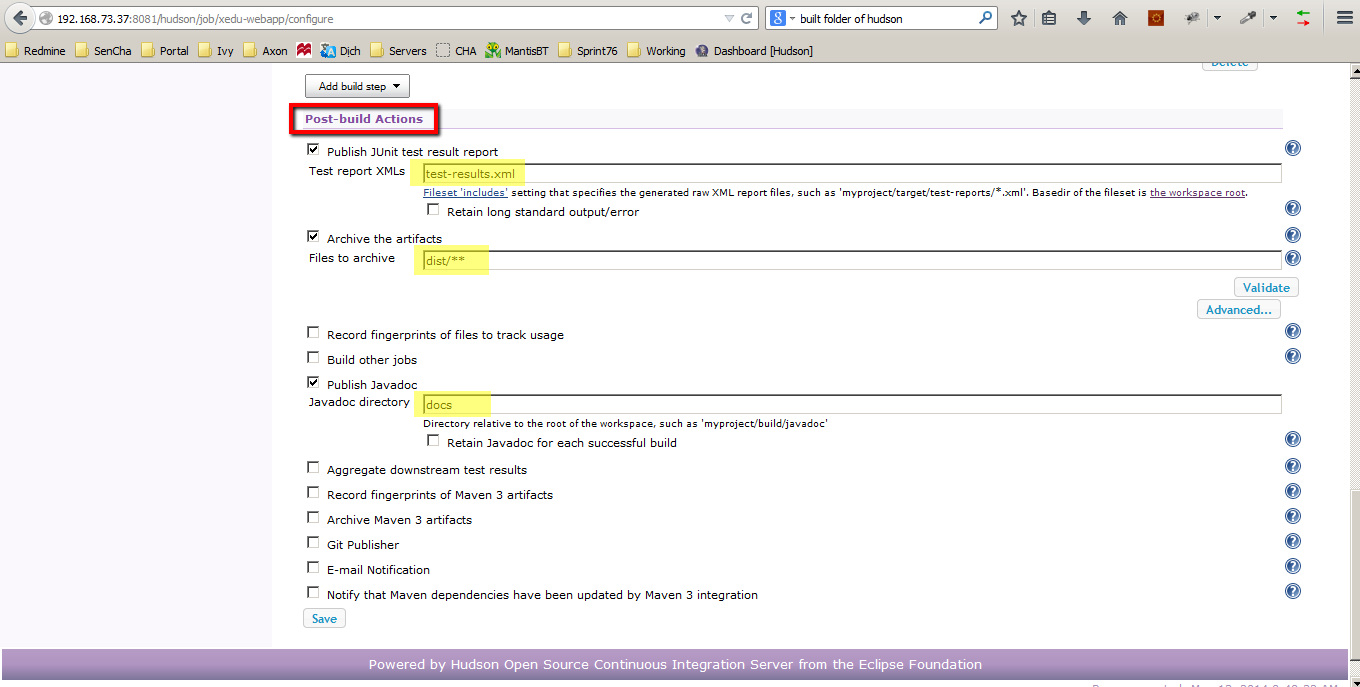
Figure 3.Bower install stuck for too long

1. Assume that your build successful.

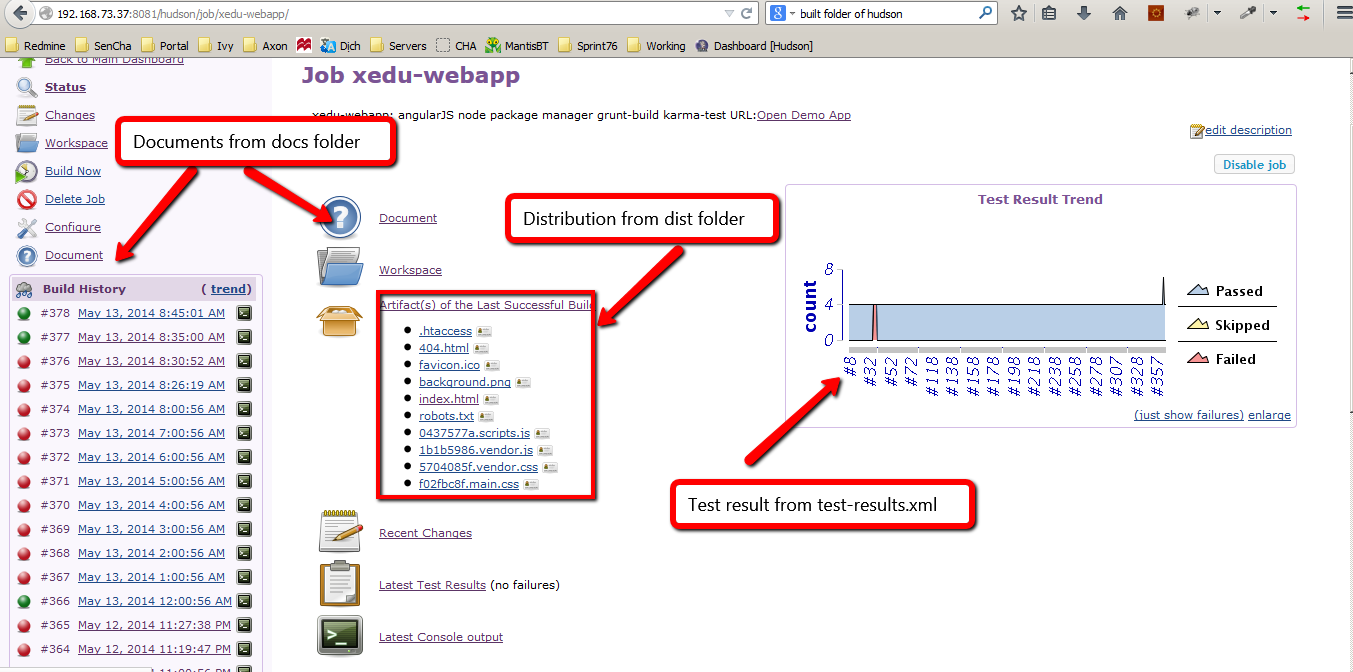
Click on **Workspace** and you will see:



Now, we will configure for Post-build actions.



Click **Save**.



More about Hudson job workspace:

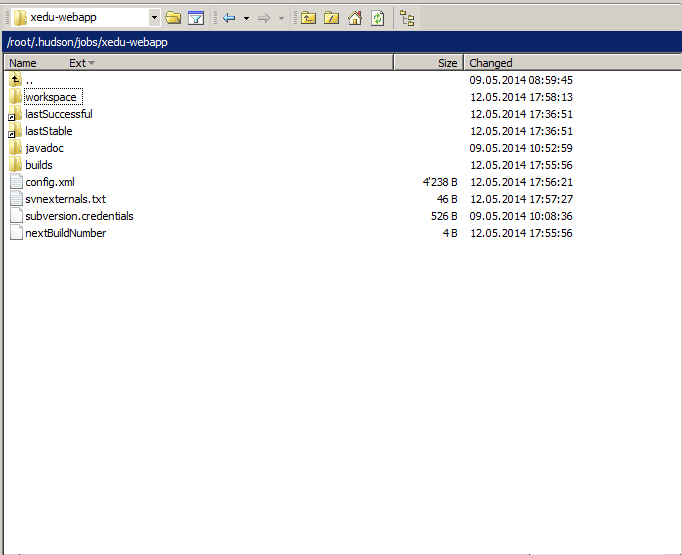


Figure 4.Workspace of job xedu-webapp in Hudson

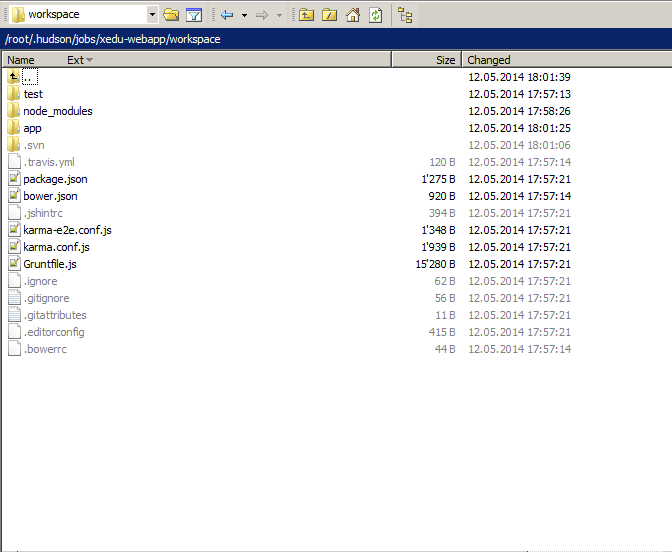


Figure 5.Workspace of xedu-webapp, this is source code get from SVN

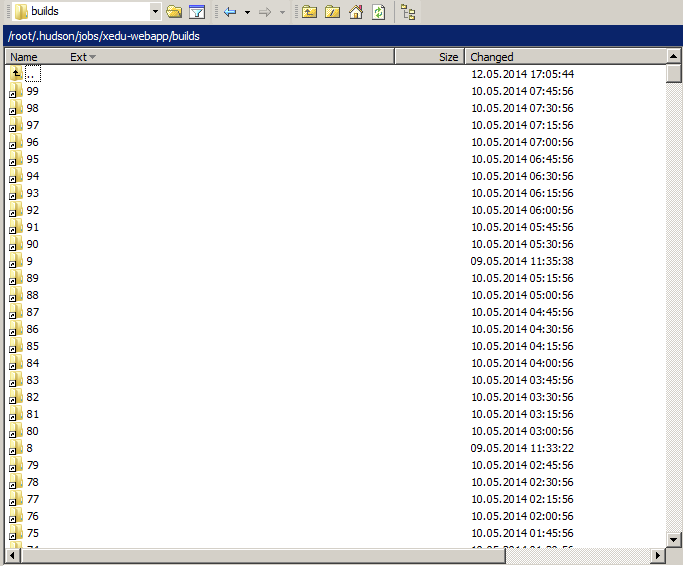
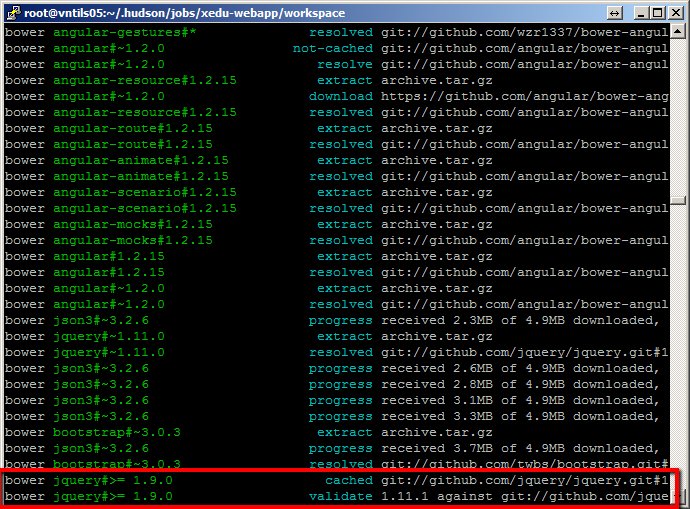


Figure 6.Builds folder, contain result of each successful build

# Remaining problems

## Bower not run correctly on Server

Bower not run with root user and stuck while get jquery.



My solution is manually copy **bower\_components** folder to server.

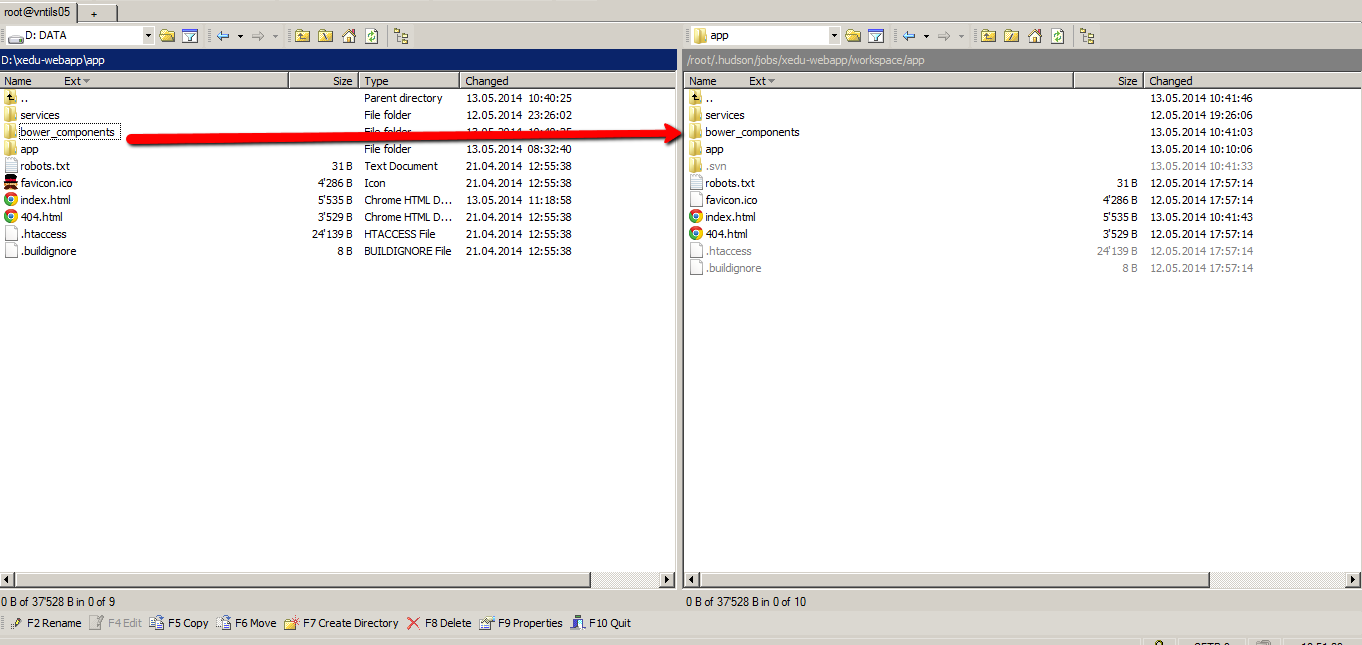


Figure 7.Use Win SCP for manually copy

And remove “**bower install**” command from build section of job configuration.

Now every time you want to update **bower\_components** you must do it manually.