BAMBOO

INSTALLATION &

LAB ASSIGNMENTS

Installation Steps:

1. **In a command prompt, run this command:**

```
java -version
```

The version of Java should be 1.8.x. Note that Bamboo requires the Java JDK to work, not the Java JRE.

Check that Windows can find Java
Bamboo uses the JAVA_HOME environment variable to find Java. To check that, in a command prompt, run:

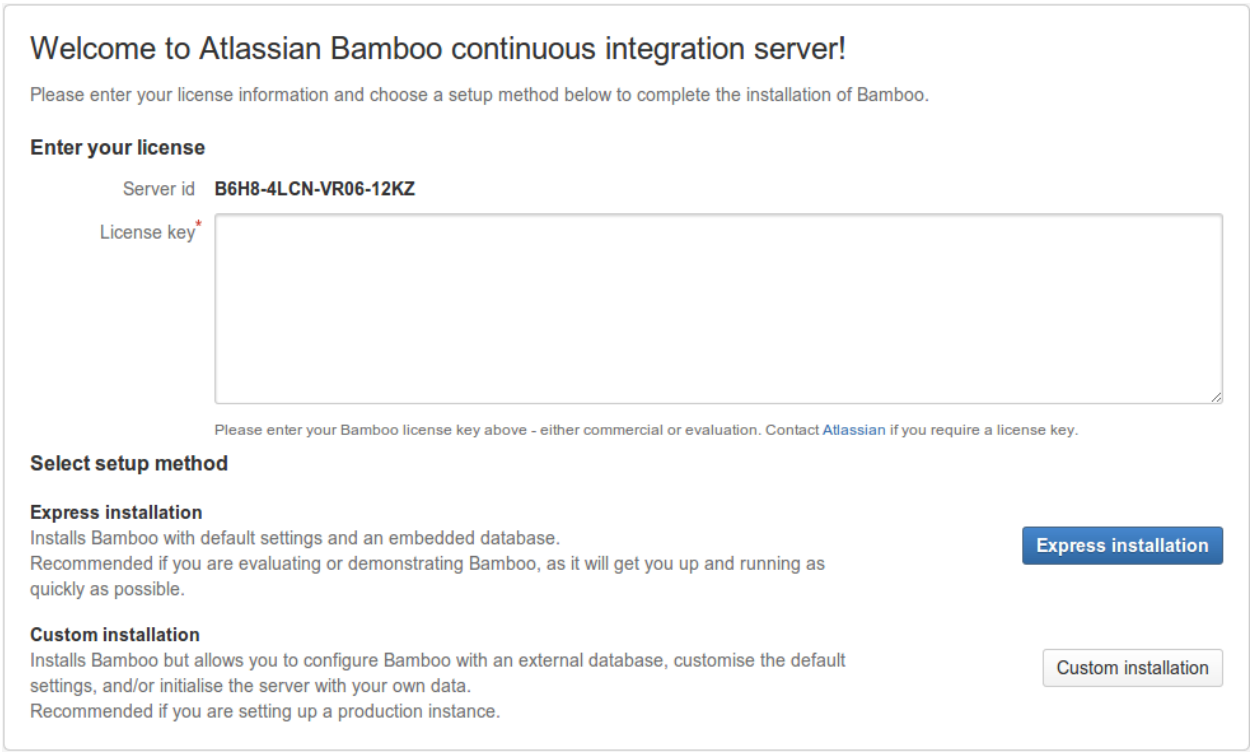
```
echo %JAVA_HOME%
```

You should see a path to the root directory of the Java installation. When running Bamboo on Windows, unlike Linux or Unix, JAVA_HOME paths with spaces are just fine.

2. **Download Bamboo**
Use the below link to download Bamboo 30 days trail version.
https://www.atlassian.com/software/bamboo/download

   a. This link will download one zip folder, unzip it you will get one exe within it.
   b. Just double click the exe to start the installation.
   c. During the installation mention Bamboo home directory to install.
   d. Simply give next to complete the installation.

3. **License Details and Setup Method**
Once the installation is completed, you can see below window prompting license details. During the trial period **Atlassian Bamboo** will provide license key to enter. You can use that key(this is one time usage, once the machine is installed with trail license, we are not allowed to re-install the trail version again in the same machine).



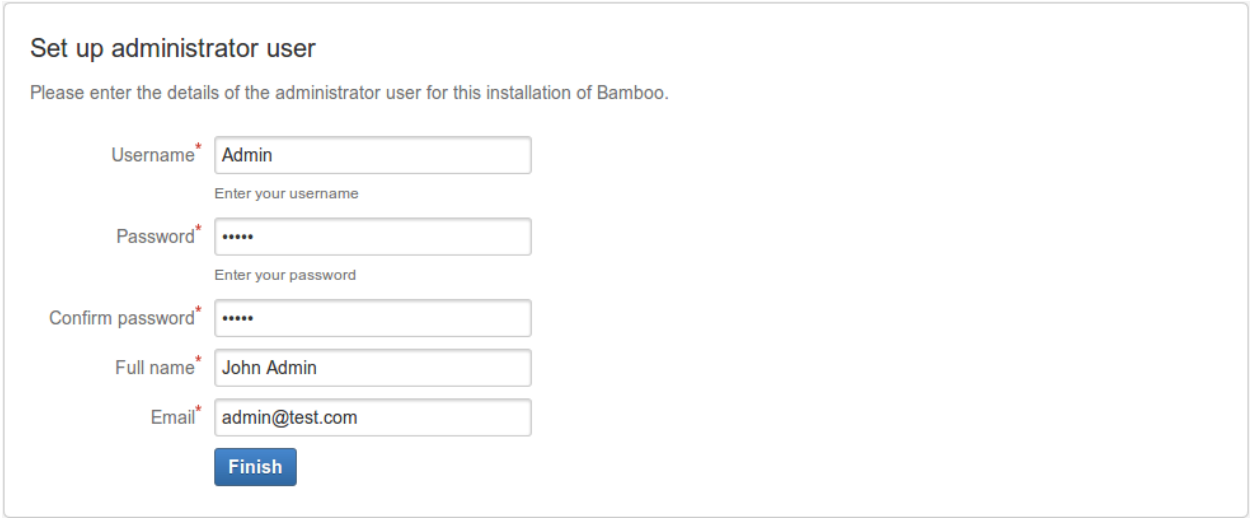
4. **Set Up Administrator User**

   The final step of the setup wizard is to enter the details of the first registered user for the Bamboo system. This user will have global administrative privileges over the entire installation of Bamboo and should not be removed.

   Once you have entered the details for your administrator user, click **Finish**. The Bamboo dashboard will be displayed.

   Congratulations, you have successfully set up Bamboo!

   Screenshot: Set Up Administrator User



5. **Congratulations you have successfully completed the installation of Bamboo.**

How to start Bamboo server once installed:

1. You can start Bamboo server by using the below.
   Start Menu → Bamboo → Start in Console.
2. Once console window will appear with log status of Bamboo. Once server started successfully, go to browser and check your server by using the below link:
   http://localhost:8085
3. By default Bamboo server will take 8085 port.
4. You can see the Bamboo dashboard where you can enter project configuration details to build your project.

Install Sonar for Bamboo:

Installing Sonar for Bamboo directly from Atlassian Marketplace

1. Click the admin drop-down on your Bamboo instance and choose **Add-ons**.
2. Click **Find new add-ons** from the left-hand side of the page.
3. Locate **Sonar for Bamboo** via search.
4. Click **Install to download** and install Sonar for Bamboo.

Installing Sonar for Bamboo by file upload

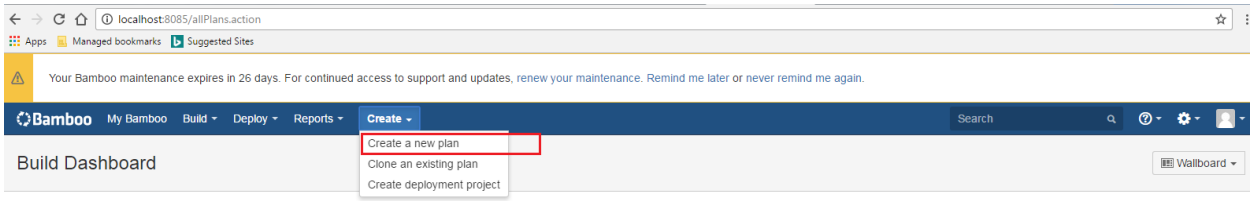
1. Download Sonar for Bamboo on the Atlassian Marketplace (Details Tab, click the download button).
2. Click the **admin** drop-down on your Bamboo instance and choose **Add-ons.**
3. Click **Manage Add-ons...**
4. Click the Upload add-on link at the top right side of the page.
5. Enter the location of **the JAR file you downloaded**.
6. Click Upload.

Lab 1:

Checkout Maven java project source code from SVN repository and build it with Bamboo Server, Once your build completed show the complete reports generated by Bamboo.

Steps:

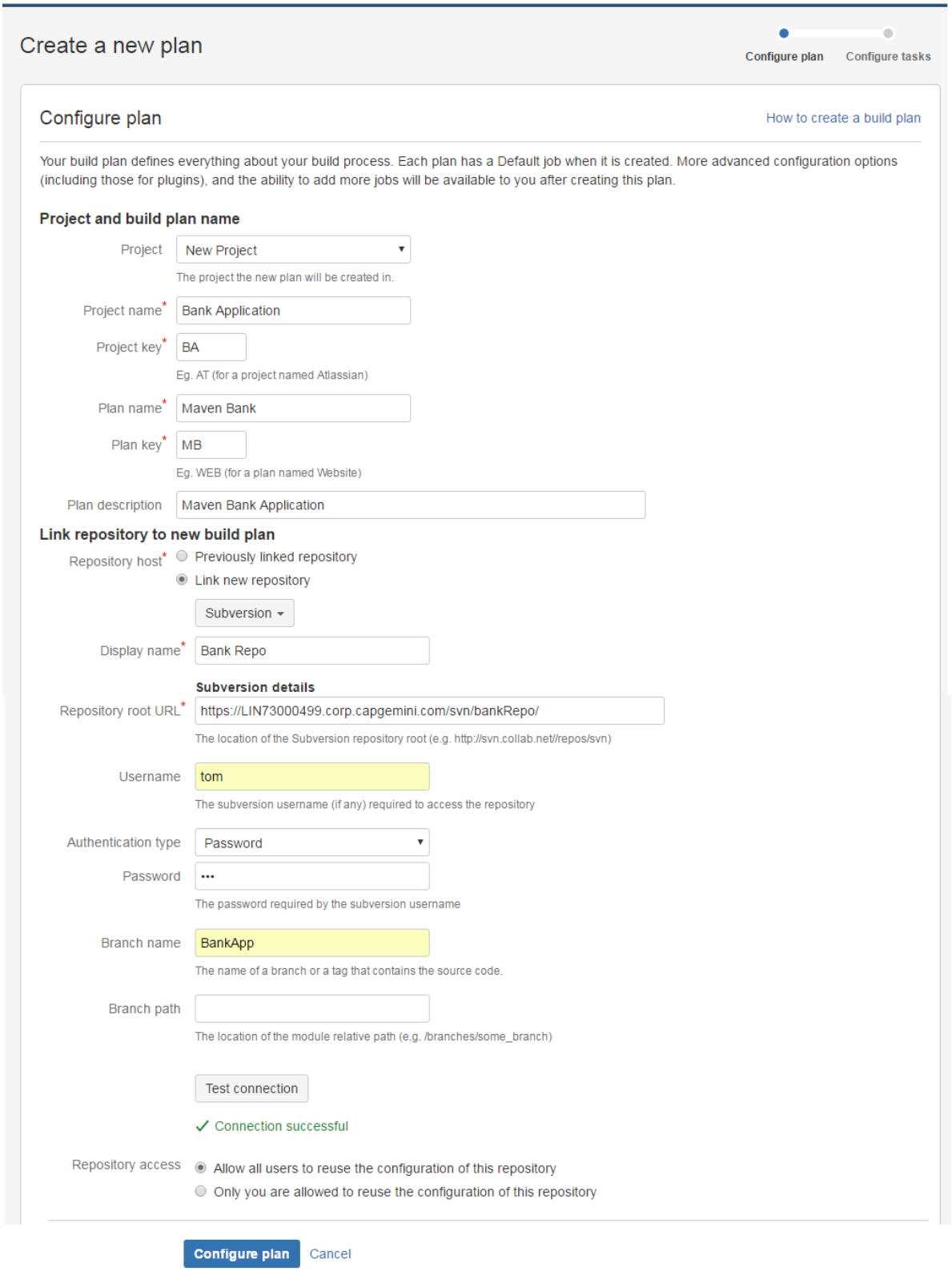
1. Once your Bamboo server is up and running (you can check http://localhost:8085) , You will get Bamboo Dashboard.
2. Click Create → Create a new plan in Bamboo Dashboard.



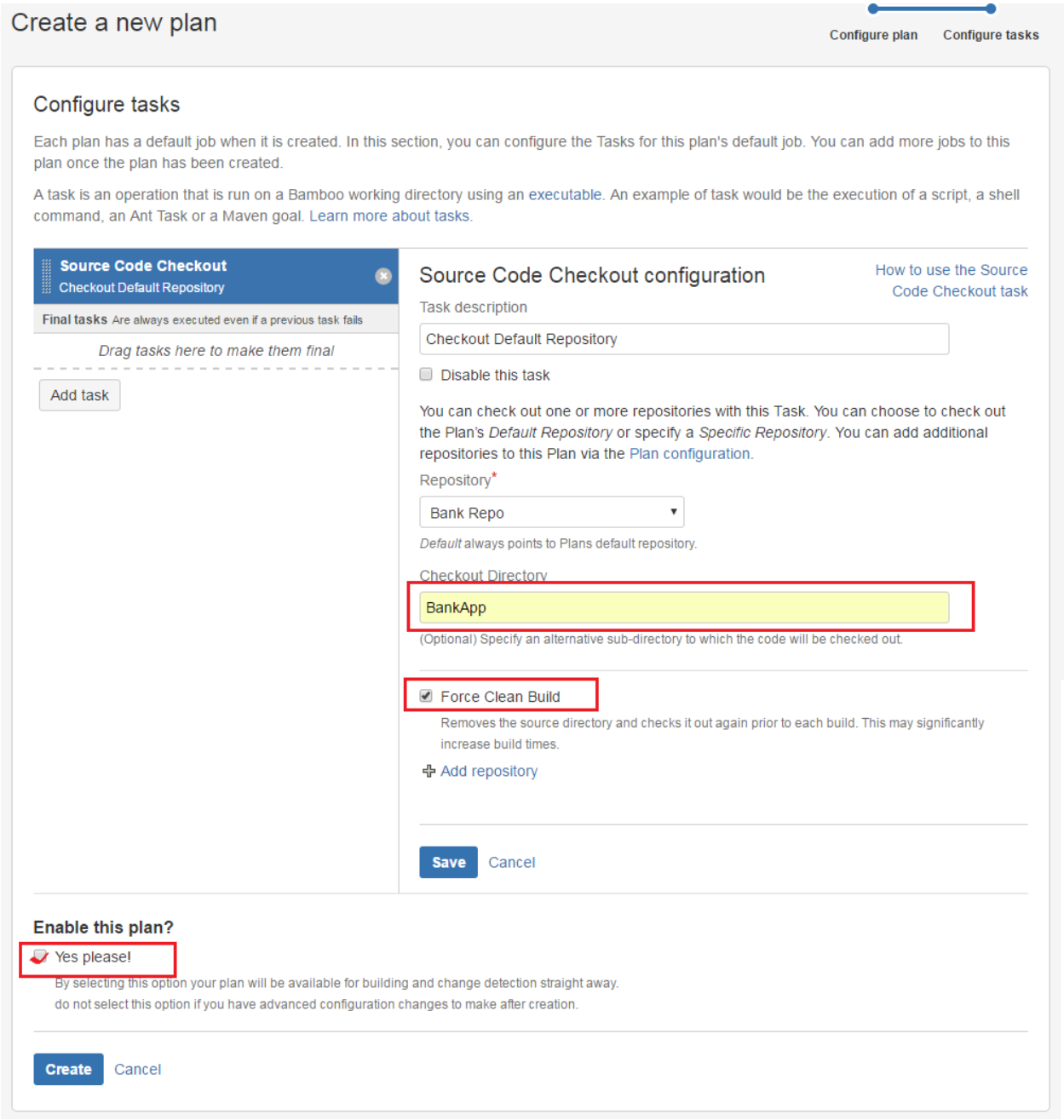
3. In **create new plan** window, please enter the below details. Then click Configure the plan.

   Note:

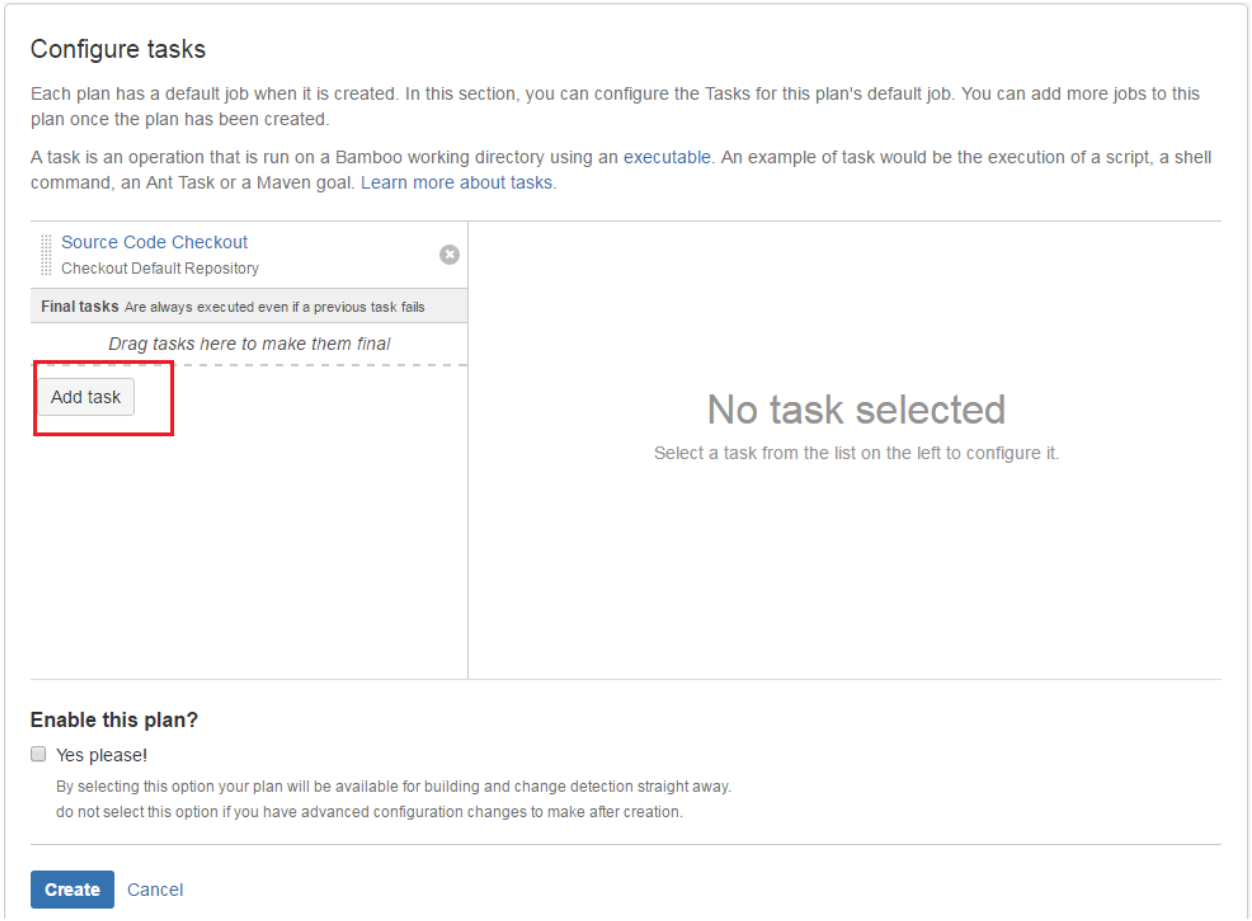
   Before create the plan commit your project with Subversion Source Control Management system. So those Bamboo servers will checkout the source code directly from SVN.


Create a new plan
Configure plan
Configure tasks
Configure plan
How to create a build plan
Your build plan defines everything about your build process. Each plan has a Default job when it is created. More advanced configuration options (including those for plugins), and the ability to add more jobs will be available to you after creating this plan.
Project and build plan name
Project
New Project
The project the new plan will be created in.
Project name*
Bank Application
Project key*
BA
Eg. AT (for a project named Atlassian)
Plan name*
Maven Bank
Plan key*
MB
Eg. WEB (for a plan named Website)
Plan description
Maven Bank Application
Link repository to new build plan
Repository host*
Previously linked repository
Link new repository
Subversion
Display name*
Bank Repo
Subversion details
Repository root URL*
https://LIN73000499.corp.capgemini.com/svn/bankRepo/
The location of the Subversion repository root (e.g. http://svn.collab.net//repos/svn)
Username
tom
The subversion username (if any) required to access the repository
Authentication type
Password
Password
The password required by the subversion username
Branch name
BankApp
The name of a branch or a tag that contains the source code.
Branch path
The location of the module relative path (e.g. /branches/some_branch)
Test connection
Connection successful
Repository access
Allow all users to reuse the configuration of this repository
Only you are allowed to reuse the configuration of this repository
Configure plan
Cancel

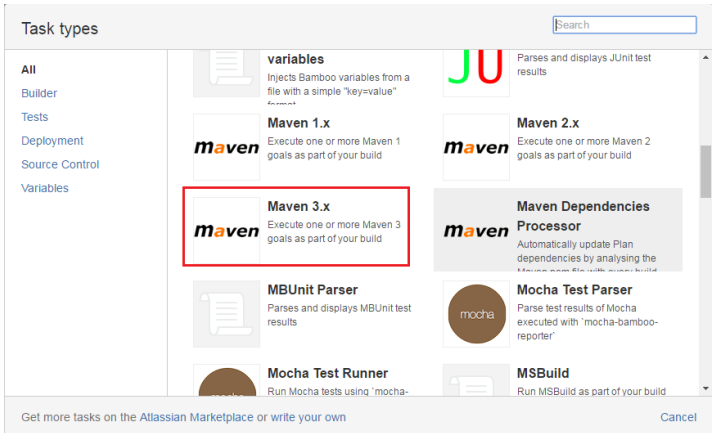
4. In Create a New plan window, change the highlighted options as per your project need, and then click Save button.



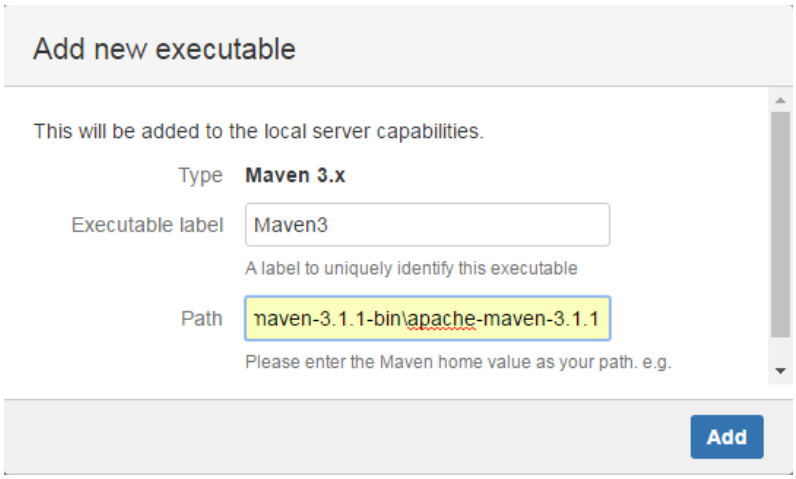
5. Click Add Task button:



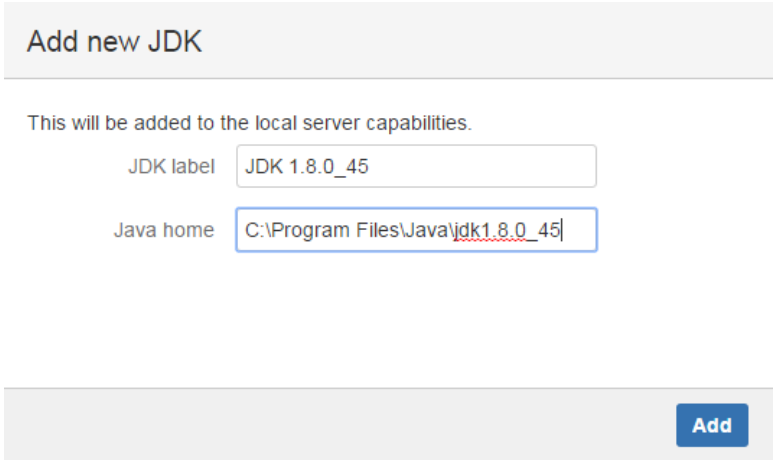
6. In the popup select Maven 3.x task type.



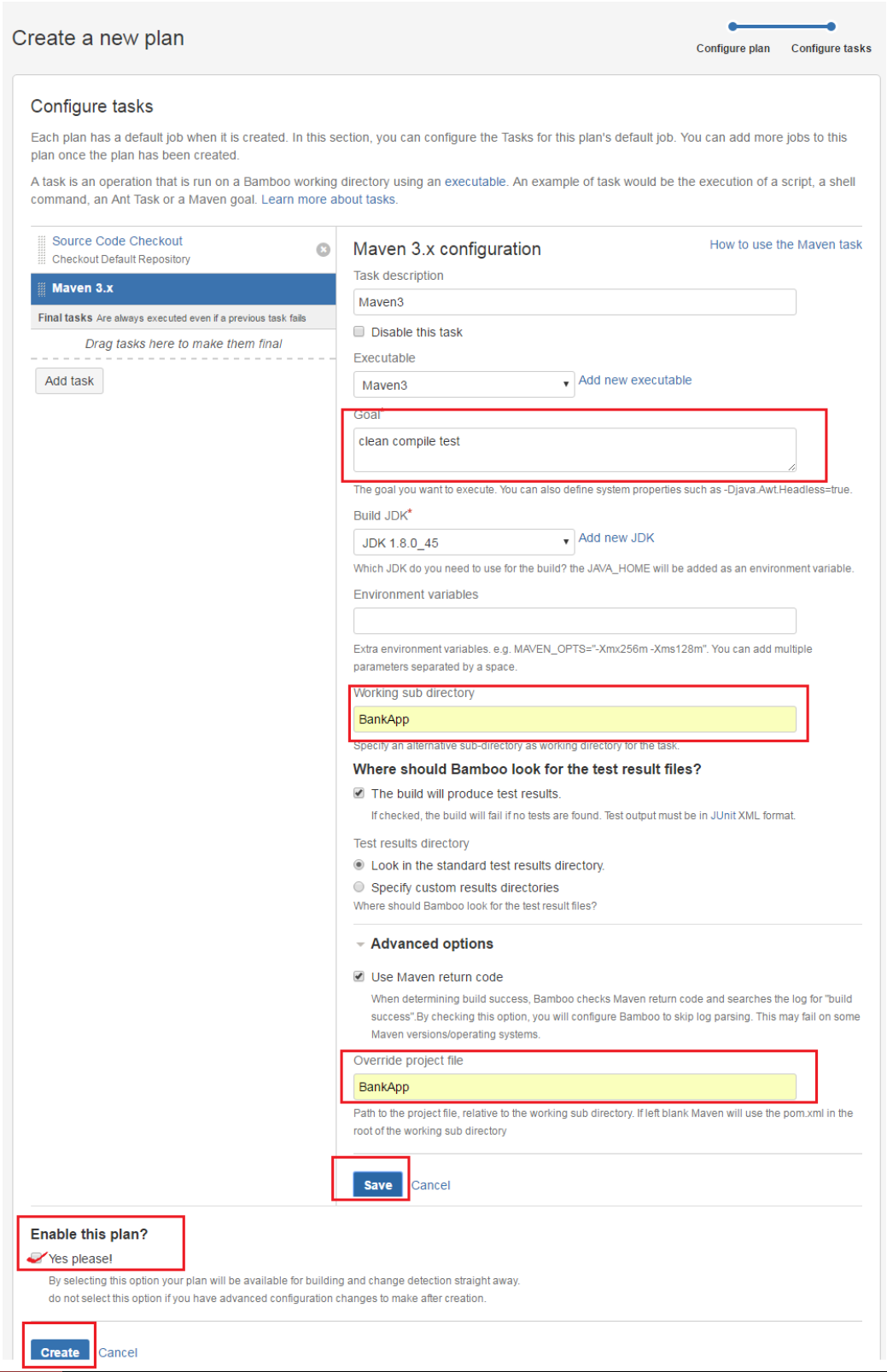
7. Click **Add new Executables**. Mention the label as **Maven3,** Add maven Home path D:\vidavid\Maven\apache-maven-3.1.1-bin\apache-maven-3.1.1. And then click Add button.



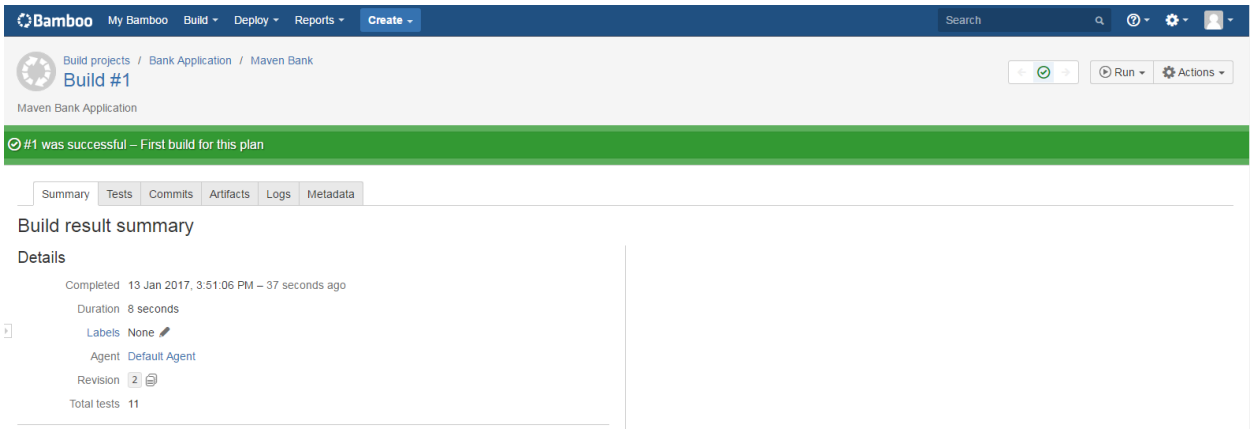
8. Click **Add new JDK.** Mention the label as JDK 1.8.0_45 and then provide Java Home path **C:\Program Files\Java\jdk1.8.0_45.** And then click Add button.



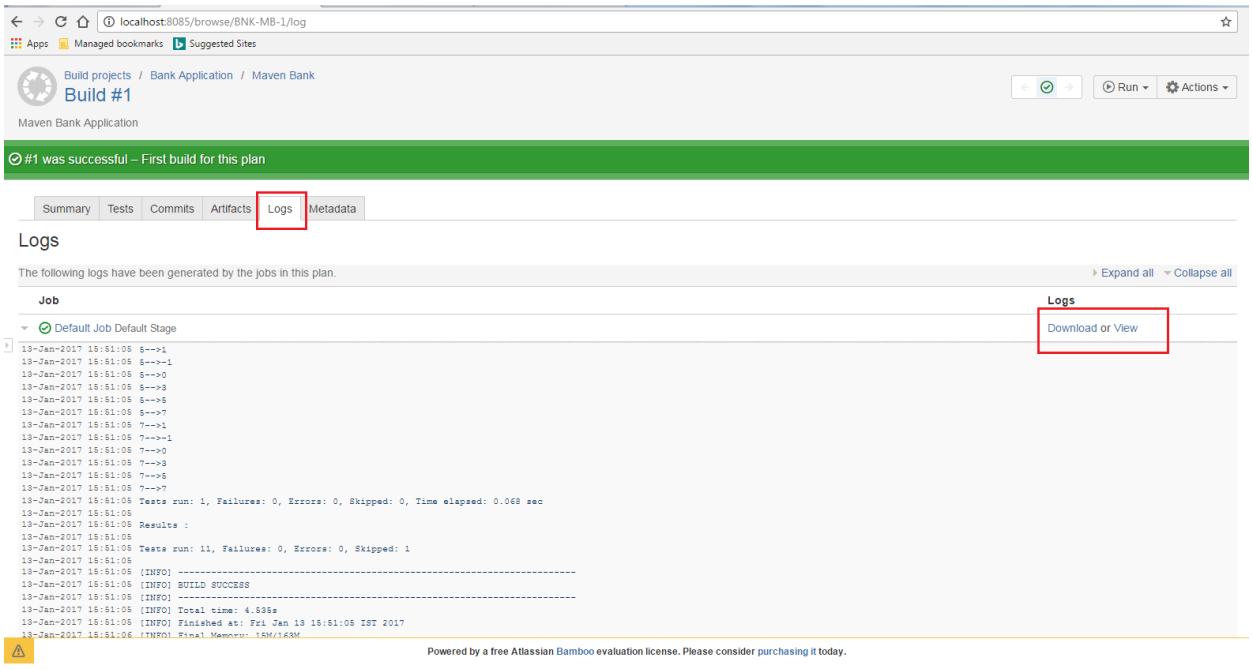
9. Fill the remaining details as highlighted below
    a. GOAL → clean compile test
    b. Working Sub Directory → BankApp (Project Root Directory under SVN)
    c. Overide Proejct File → BankApp (Project Root Directory under SVN)
    d. Enable this plan. And and click create button.


Create a new plan
Configure plan
Configure tasks
Configure tasks
Each plan has a default job when it is created. In this section, you can configure the Tasks for this plan's default job. You can add more jobs to this plan once the plan has been created.
A task is an operation that is run on a Bamboo working directory using an executable. An example of task would be the execution of a script, a shell command, an Ant Task or a Maven goal. Learn more about tasks.
Source Code Checkout
Checkout Default Repository
Maven 3.x
Final tasks Are always executed even if a previous task fails
Drag tasks here to make them final
Add task
Maven 3.x configuration
How to use the Maven task
Task description
Maven3
Disable this task
Executable
Maven3
Add new executable
Goal*
clean compile test
The goal you want to execute. You can also define system properties such as -Djava.Awt.Headless=true.
Build JDK*
JDK 1.8.0_45
Add new JDK
Which JDK do you need to use for the build? the JAVA_HOME will be added as an environment variable.
Environment variables
Extra environment variables. e.g. MAVEN_OPTS="-Xmx256m -Xms128m". You can add multiple parameters separated by a space.
Working sub directory
BankApp
Specify an alternative sub-directory as working directory for the task.
Where should Bamboo look for the test result files?
The build will produce test results.
If checked, the build will fail if no tests are found. Test output must be in JUnit XML format.
Test results directory
Look in the standard test results directory.
Specify custom results directories
Where should Bamboo look for the test result files?
Advanced options
Use Maven return code
When determining build success, Bamboo checks Maven return code and searches the log for "build success".By checking this option, you will configure Bamboo to skip log parsing. This may fail on some Maven versions/operating systems.
Override project file
BankApp
Path to the project file, relative to the working sub directory. If left blank Maven will use the pom.xml in the root of the working sub directory
Save
Cancel
Enable this plan?
Yes please!
By selecting this option your plan will be available for building and change detection straight away.
do not select this option if you have advanced configuration changes to make after creation.
Create
Cancel

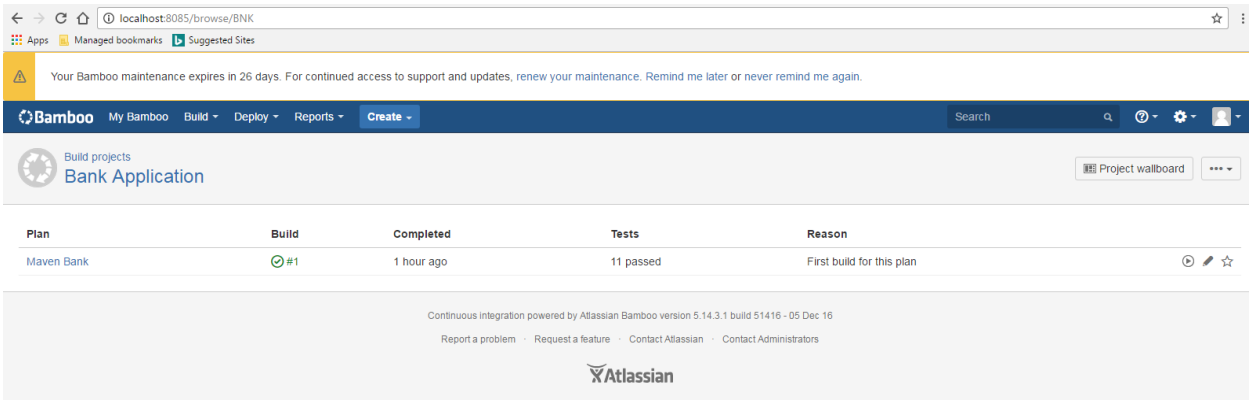
10. Once you click create button, build process will start automatically. You can see the build result as shown below:



11. If you want to check the log report, under the logs either via download/view you can see the complete log report generated by Bamboo.



13. Now if you navigate to dashboard, it will show the no of build happen under the project.



Learning:

We have learnt how to build Java project which is build by Maven has been build successfully in Bamboo server.

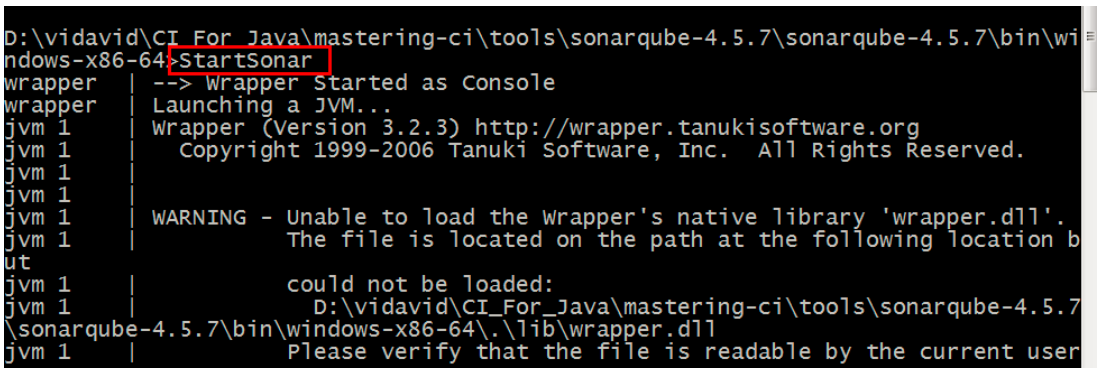
Congratulation! You have successfully completed the Lab! ☺

Lab 2:

Checkout Maven java project source code from SVN repository and build it with Bamboo Server. Integrate SonarQube Server to check code quality once your build successfully completed.

SonarQube in your local Machine:

1. Download **SonarQube** from the below link
   a. http://www.sonarqube.org/downloads/
2. Locate the bin directory, choose the file which is appropriate for your machine (Example if you are using windows 64 bit, then open windows-x86-64)
3. Then enter StartSonar command in your command prompt.
4. It will start SonarQube Server as mentioned below:



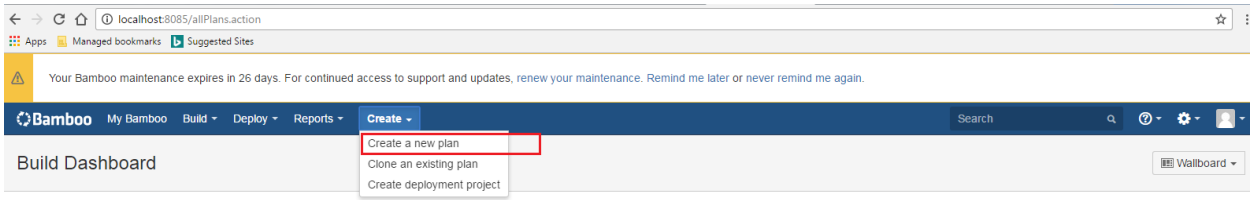
By default SonarQube will take 9000 port, If you wish to see SonarQube started properly you can open http://localhost:9000

Steps:

1. To check code quality with Sonar Server, add the below plugin in pom.xml file in your Maven project. And commit the source code.

```
<plugin>

    <groupId>org.sonarsource.scanner.maven</groupId>

    <artifactId>sonar-maven-plugin</artifactId>

    <version>3.1.1</version>

  </plugin>
```

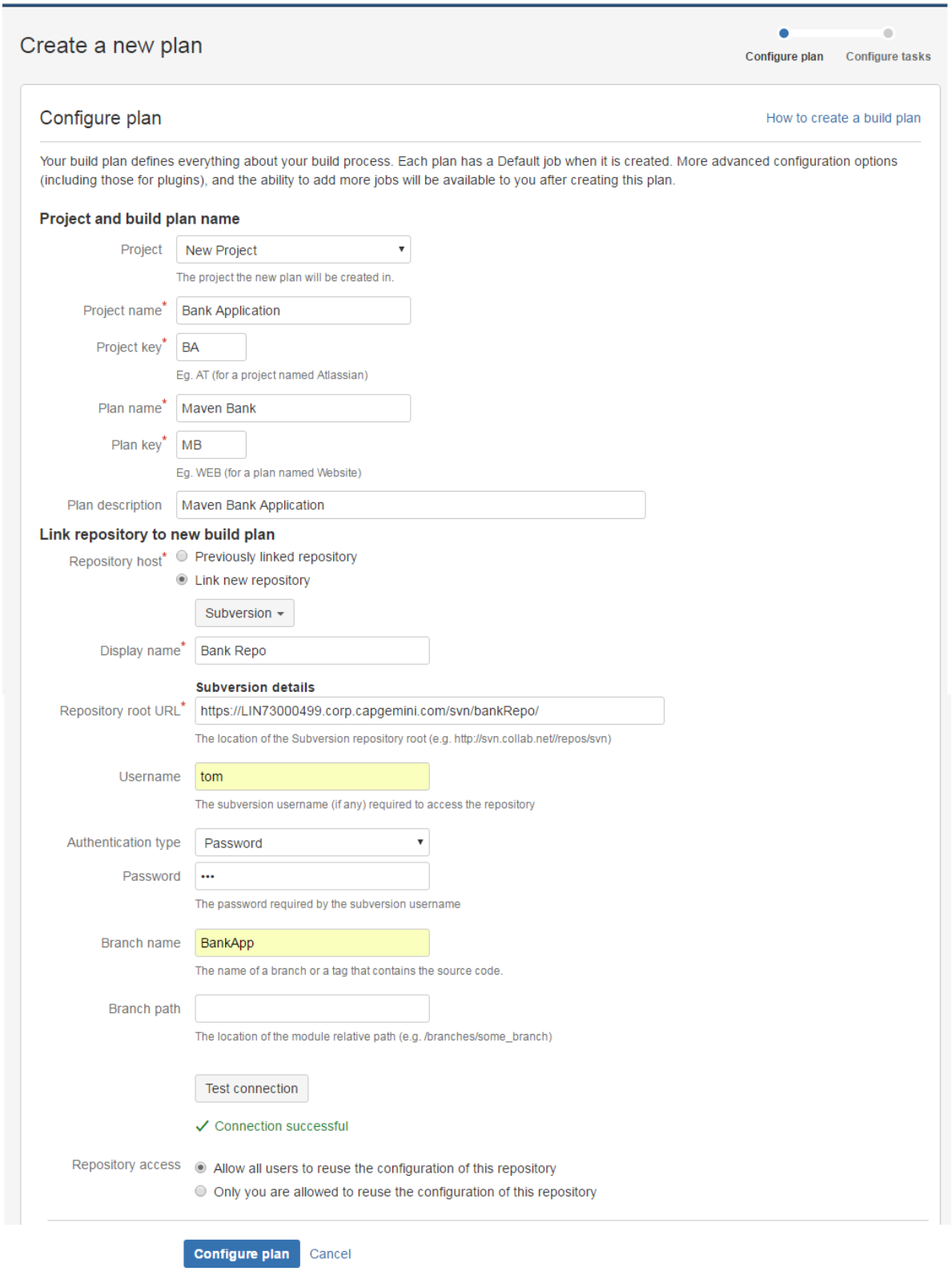
2. Once your Bamboo server is up and running (you can check http://localhost:8085) , You will get Bamboo Dashboard.
3. Click Create → Create a new plan in Bamboo Dashboard.



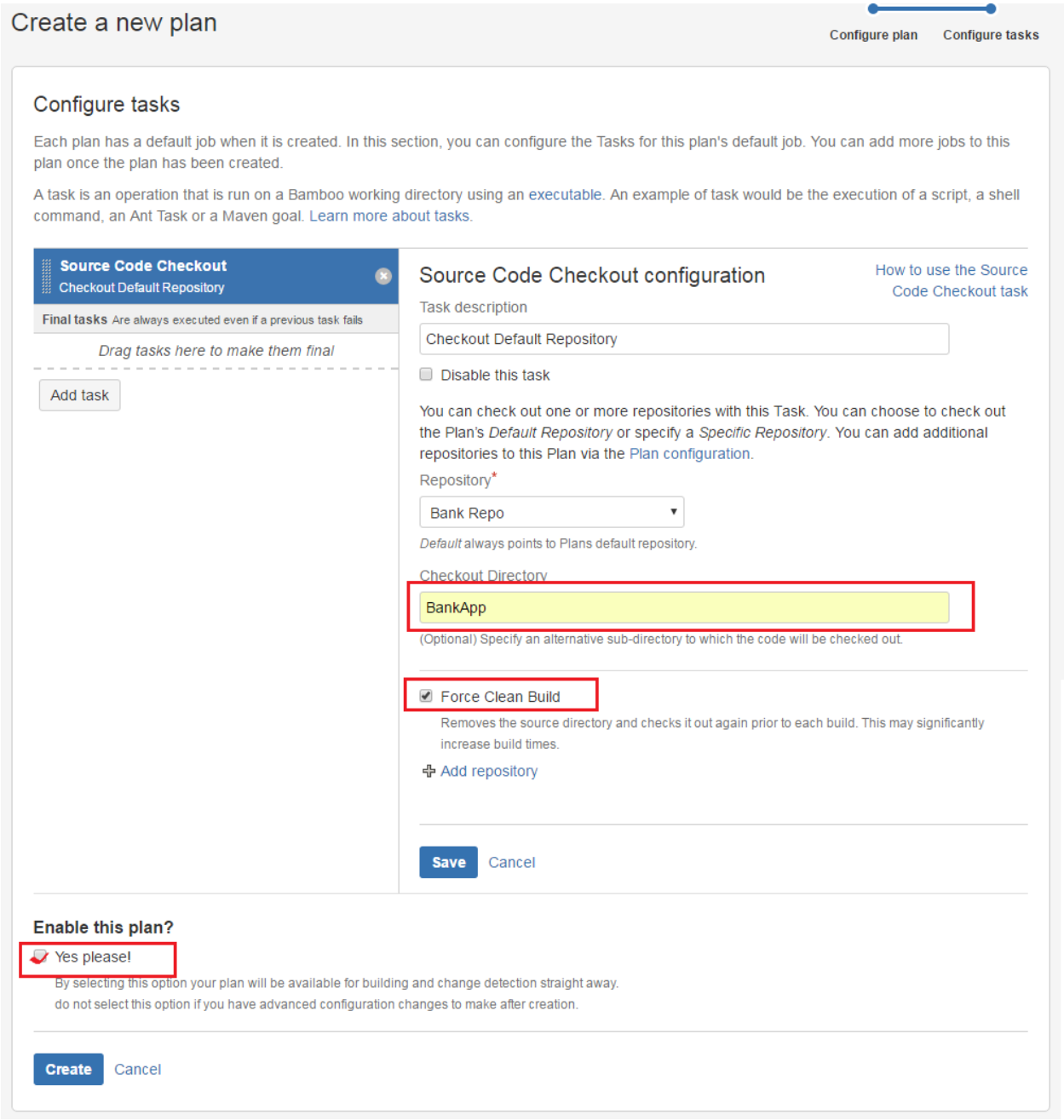
4. In **create new plan** window, please enter the below details. Then click Configure the plan.

Note:

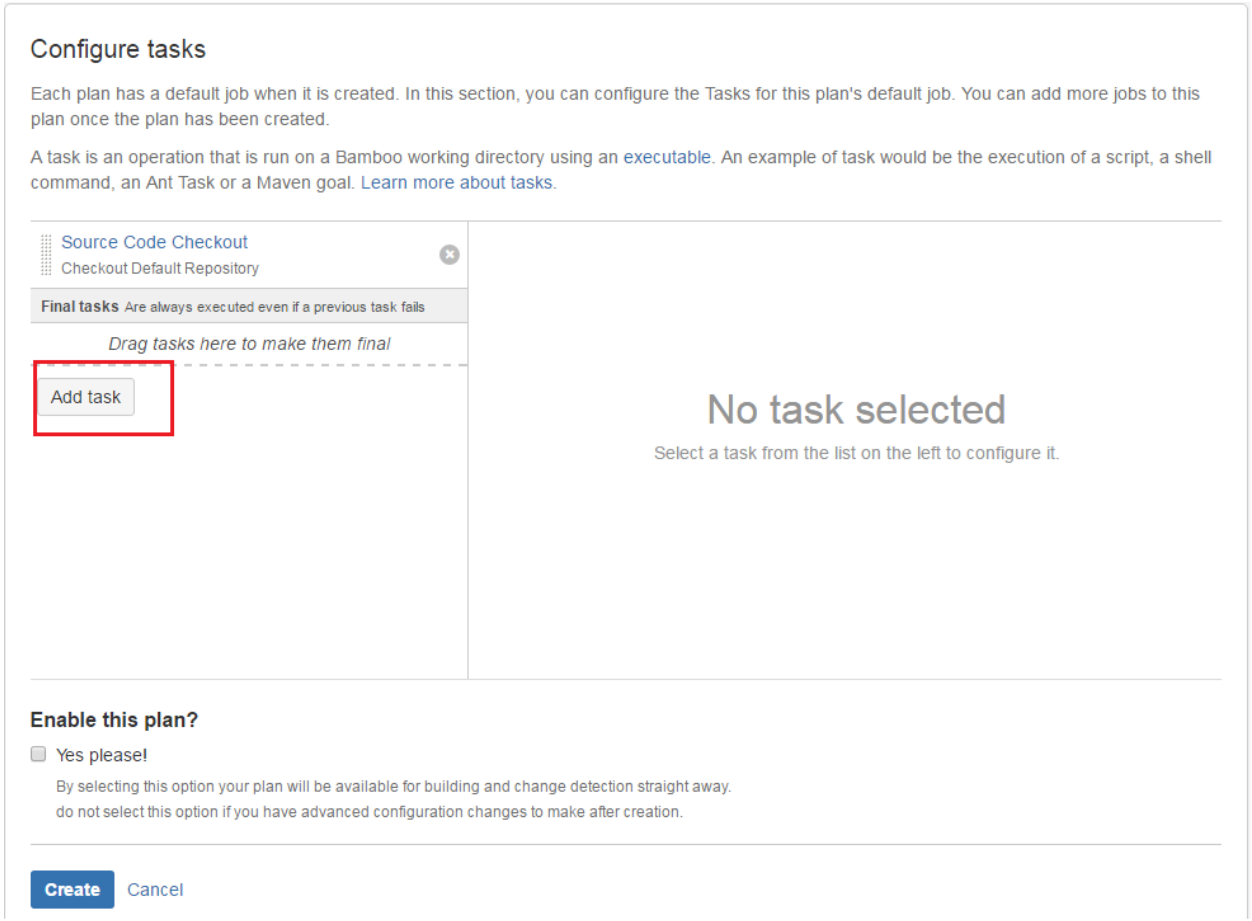
Before create the plan commit your project with Subversion Source Control Management system. So those Bamboo servers will checkout the source code directly from SVN.


Create a new plan
Configure plan
Configure tasks
Configure plan
How to create a build plan
Your build plan defines everything about your build process. Each plan has a Default job when it is created. More advanced configuration options (including those for plugins), and the ability to add more jobs will be available to you after creating this plan.
Project and build plan name
Project
New Project
The project the new plan will be created in.
Project name*
Bank Application
Project key*
BA
Eg. AT (for a project named Atlassian)
Plan name*
Maven Bank
Plan key*
MB
Eg. WEB (for a plan named Website)
Plan description
Maven Bank Application
Link repository to new build plan
Repository host*
Previously linked repository
Link new repository
Subversion
Display name*
Bank Repo
Subversion details
Repository root URL*
https://LIN73000499.corp.capgemini.com/svn/bankRepo/
The location of the Subversion repository root (e.g. http://svn.collab.net//repos/svn)
Username
tom
The subversion username (if any) required to access the repository
Authentication type
Password
Password
The password required by the subversion username
Branch name
BankApp
The name of a branch or a tag that contains the source code.
Branch path
The location of the module relative path (e.g. /branches/some_branch)
Test connection
Connection successful
Repository access
Allow all users to reuse the configuration of this repository
Only you are allowed to reuse the configuration of this repository
Configure plan
Cancel

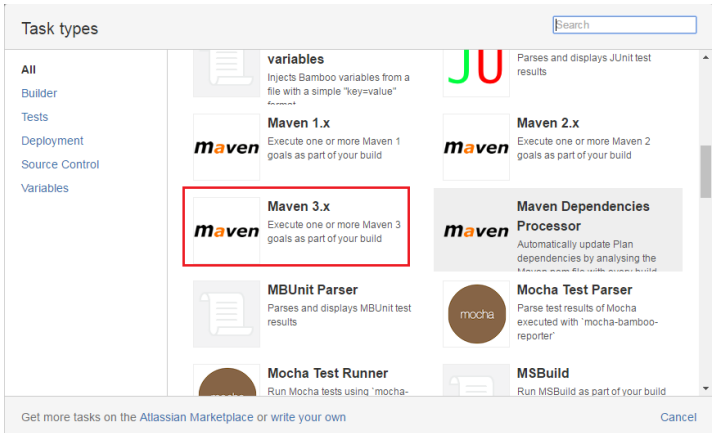
5. In Create a New plan window, change the highlighted options as per your project need, and then click Save button.



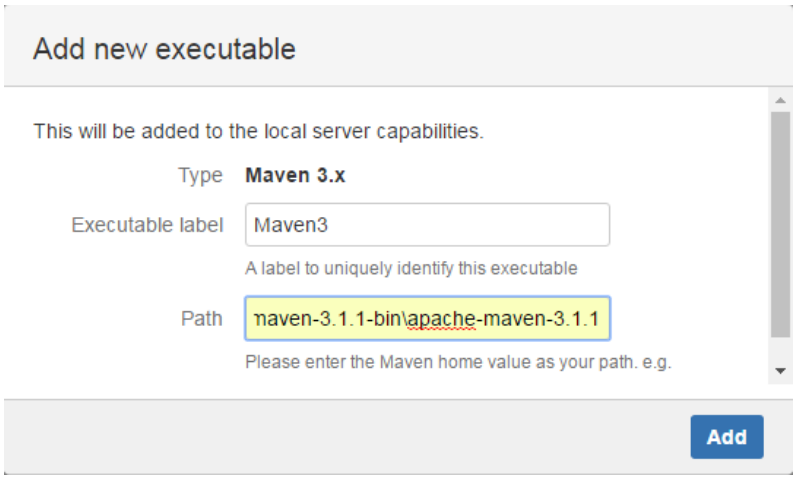
6. Click Add Task button:



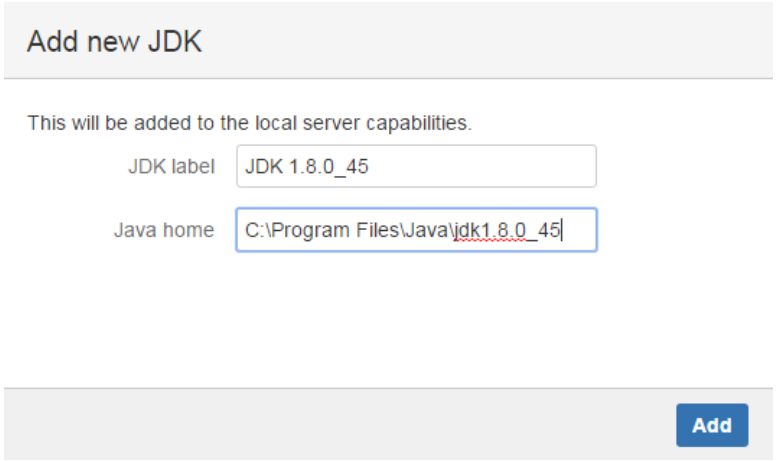
7. In the popup select Maven 3.x task type.



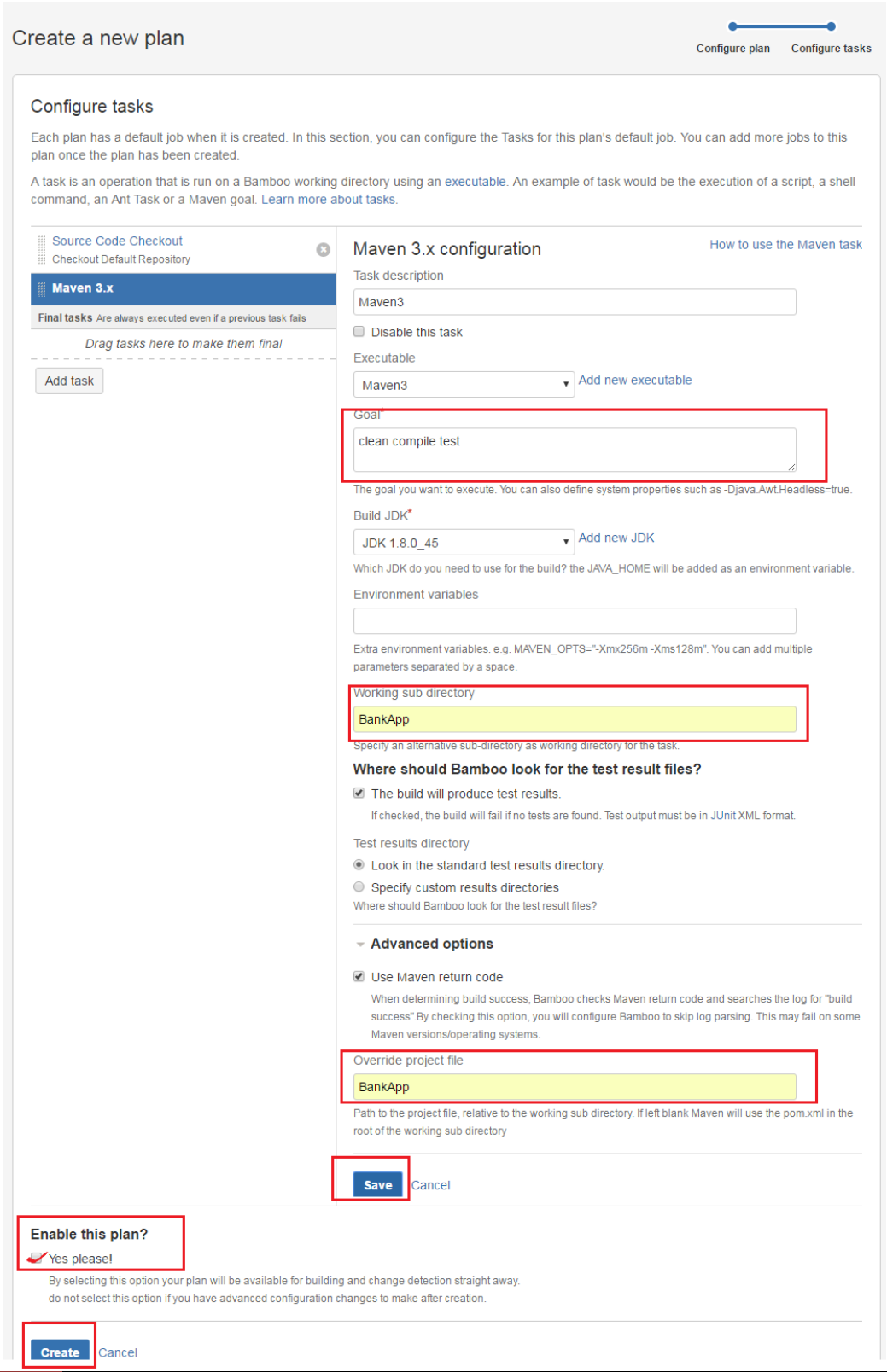
8. Click **Add new Executables**. Mention the label as **Maven3,** Add maven Home path D:\vidavid\Maven\apache-maven-3.1.1-bin\apache-maven-3.1.1. And then click Add button.



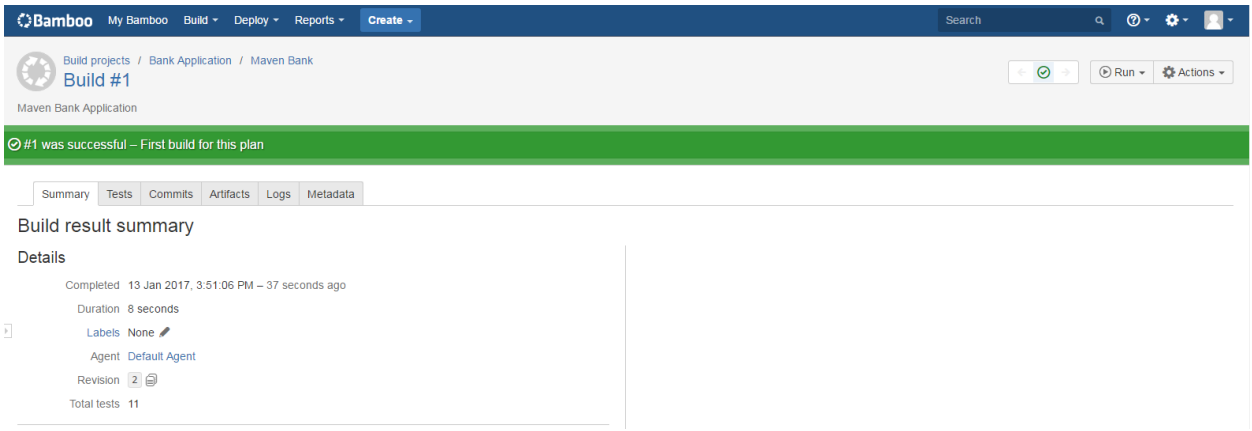
9. Click **Add new JDK.** Mention the label as JDK 1.8.0_45 and then provide Java Home path **C:\Program Files\Java\jdk1.8.0_45.** And then click Add button.



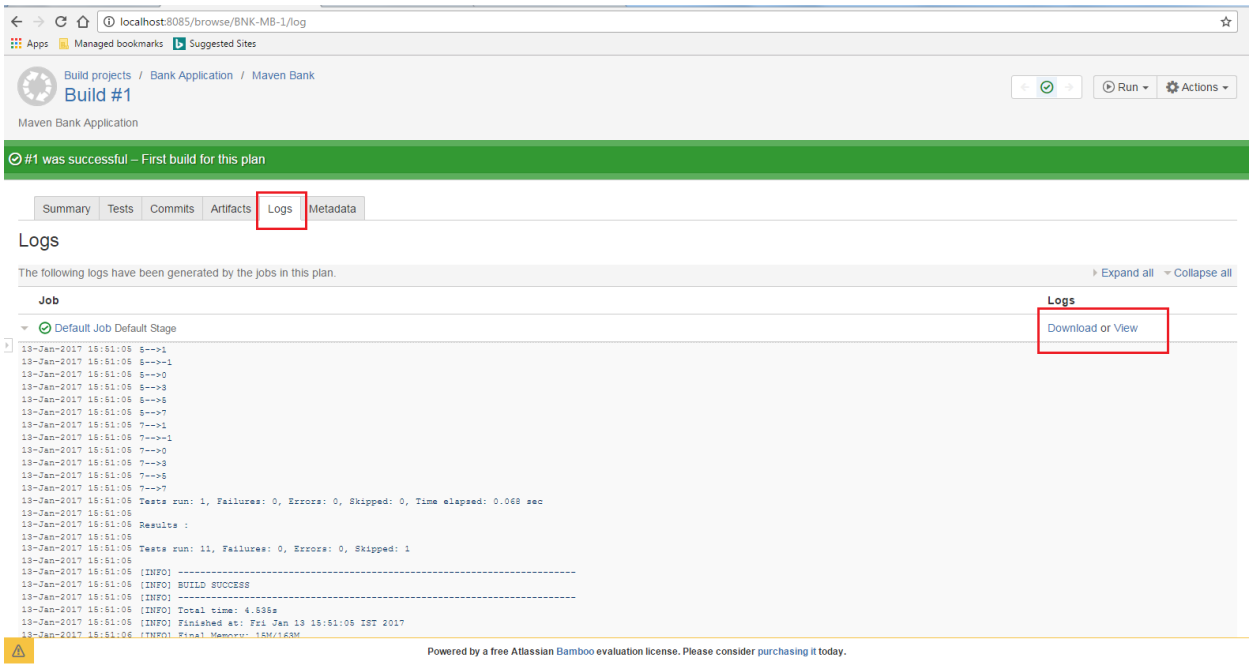
10. Fill the remaining details as highlighted below
    a. **GOAL → clean install test sonar:sonar**
    b. Working Sub Directory → BankApp (Project Root Directory under SVN)
    c. Overide Proejct File → BankApp (Project Root Directory under SVN)
    d. Enable this plan. And and click create button.


Create a new plan
Configure plan
Configure tasks
Configure tasks
Each plan has a default job when it is created. In this section, you can configure the Tasks for this plan's default job. You can add more jobs to this plan once the plan has been created.
A task is an operation that is run on a Bamboo working directory using an executable. An example of task would be the execution of a script, a shell command, an Ant Task or a Maven goal. Learn more about tasks.
Source Code Checkout
Checkout Default Repository
Maven 3.x
Final tasks Are always executed even if a previous task fails
Drag tasks here to make them final
Add task
Maven 3.x configuration
How to use the Maven task
Task description
Maven3
Disable this task
Executable
Maven3
Add new executable
Goal*
clean compile test
The goal you want to execute. You can also define system properties such as -Djava.Awt.Headless=true.
Build JDK*
JDK 1.8.0_45
Add new JDK
Which JDK do you need to use for the build? the JAVA_HOME will be added as an environment variable.
Environment variables
Extra environment variables. e.g. MAVEN_OPTS="-Xmx256m -Xms128m". You can add multiple parameters separated by a space.
Working sub directory
BankApp
Specify an alternative sub-directory as working directory for the task.
Where should Bamboo look for the test result files?
The build will produce test results.
If checked, the build will fail if no tests are found. Test output must be in JUnit XML format.
Test results directory
Look in the standard test results directory.
Specify custom results directories
Where should Bamboo look for the test result files?
Advanced options
Use Maven return code
When determining build success, Bamboo checks Maven return code and searches the log for "build success".By checking this option, you will configure Bamboo to skip log parsing. This may fail on some Maven versions/operating systems.
Override project file
BankApp
Path to the project file, relative to the working sub directory. If left blank Maven will use the pom.xml in the root of the working sub directory
Save
Cancel
Enable this plan?
Yes please!
By selecting this option your plan will be available for building and change detection straight away.
do not select this option if you have advanced configuration changes to make after creation.
Create
Cancel

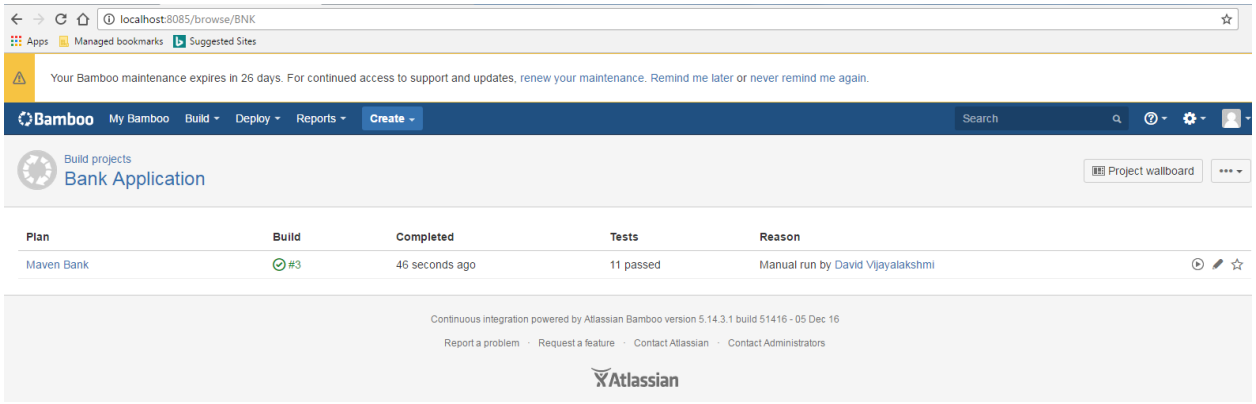
11. Once you click create button, build process will start automatically. You can see the build result as shown below:



12. If you want to check the log report, under the logs either via download/view you can see the complete log report generated by Bamboo.

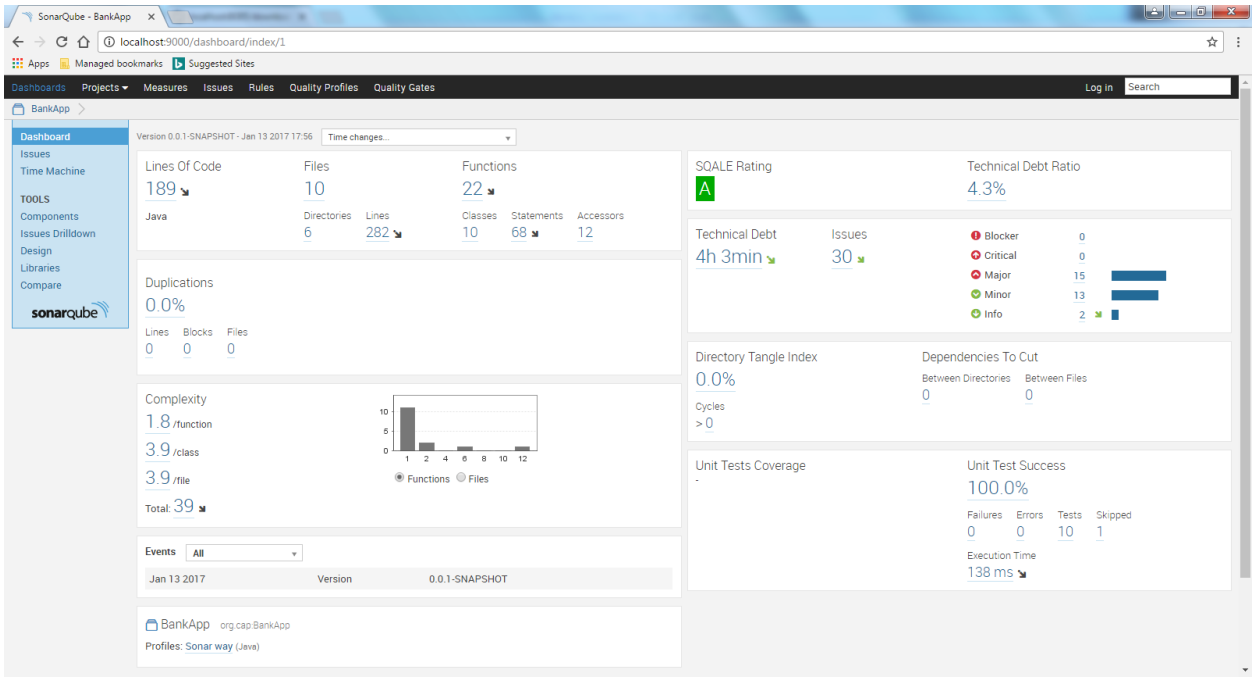


14. Now if you navigate to dashboard, it will show the no of build happen under the project.



15. Once the build completed successfully, check the below URL, you can see the complete quality check details of your code.

http://localhost:9000/dashboard/index/org.cap:BankApp



Learning:

We have learnt how to build Java project which is build by Maven in Bamboo server. And how to analyze the code in SonarQube server.

Congratulation! You have successfully completed the Lab! ☺