

# **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 trial 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).

## Welcome to Atlassian Bamboo continuous integration server!

Please enter your license information and choose a setup method below to complete the installation of Bamboo.

### Enter your license

Server id **B6H8-4LCN-VR06-12KZ**

License key\*

Please enter your Bamboo license key above - either commercial or evaluation. Contact [Atlassian](#) if you require a license key.

### Select setup method

#### Express installation

Installs Bamboo with default settings and an embedded database.

Recommended if you are evaluating or demonstrating Bamboo, as it will get you up and running as quickly as possible.

Express installation

#### Custom installation

Installs Bamboo but allows you to configure Bamboo with an external database, customise the default settings, and/or initialise the server with your own data.

Recommended if you are setting up a production instance.

Custom installation

## 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*

### Set up administrator user

Please enter the details of the administrator user for this installation of Bamboo.

Username\*

Enter your username

Password\*

Enter your password

Confirm password\*

Full name\*

Email\*

**Finish**

**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

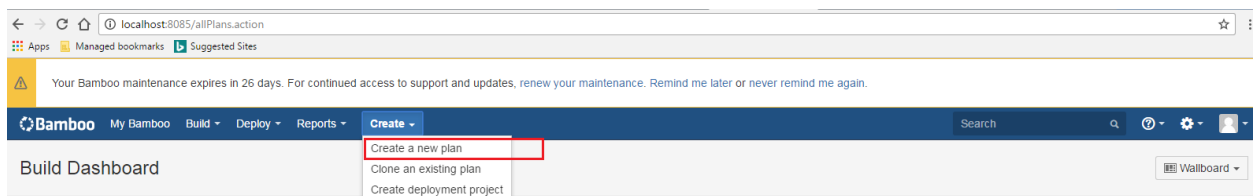
1. Download Sonar for Bamboo on the Atlassian Marketplace (Details Tab, click the download button).
2. Click the **admin** drop-down on your Bamboo instance and choose **Add-ons**.
3. Click **Manage Add-ons...**
4. Click the Upload add-on link at the top right side of the page.
5. Enter the location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

The project the new plan will be created in.

Project name\*

Project key\*

Eg. AT (for a project named Atlassian)

Plan name\*

Plan key\*

Eg. WEB (for a plan named Website)

Plan description

#### Link repository to new build plan

Repository host\* ☐ Previously linked repository

☒ Link new repository

Display name\*

#### Subversion details

Repository root URL\*

The location of the Subversion repository root (e.g. http://svn.collab.net/repos/svn)

Username

The subversion username (if any) required to access the repository

Authentication type

Password

The password required by the subversion username

Branch name

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)

✓ 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

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

Final tasks Are always executed even if a previous task fails

Drag tasks here to make them final

Add task

### Source Code Checkout configuration

How to use the Source Code Checkout task

Task description

Checkout Default Repository

☐ Disable this task

You can check out one or more repositories with this Task. You can choose to check out the Plan's *Default Repository* or specify a *Specific Repository*. You can add additional repositories to this Plan via the [Plan configuration](#).

Repository\*

Bank Repo

Default always points to Plans default repository.

Checkout Directory

BankApp

(Optional) Specify an alternative sub-directory to which the code will be checked out

☒ Force Clean Build

Removes the source directory and checks it out again prior to each build. This may significantly increase build times.

+ Add repository

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



## 5. Click Add Task button:

### 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

Final tasks Are always executed even if a previous task fails

Drag tasks here to make them final

Add task

No task selected

Select a task from the list on the left to configure it.

### 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

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

### Task types

All

Builder

Tests

Deployment

Source Control

Variables

variables

Injects Bamboo variables from a file with a simple "key=value"

Maven 1.x

Execute one or more Maven 1 goals as part of your build

Maven 3.x

Execute one or more Maven 3 goals as part of your build

MUnit Parser

Parses and displays MUnit test results

Mocha Test Runner

Run Mocha tests using "mocha-

Maven 2.x

Execute one or more Maven 2 goals as part of your build

Maven Dependencies Processor

Automatically update Plan dependencies by analysing the

Mocha Test Parser

Parse test results of Mocha executed with "mocha-bamboo-reporter"

MSBuild

Run MSBuild as part of your build

Get more tasks on the Atlassian Marketplace or write your own

Cancel

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.

### Add new executable

This will be added to the local server capabilities.

Type **Maven 3.x**

Executable label

A label to uniquely identify this executable

Path

Please enter the Maven home value as your path. e.g.

**Add**

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.

### Add new JDK

This will be added to the local server capabilities.

JDK label

Java home

**Add**

9. Fill the remaining details as highlighted below
  - a. GOAL → clean compile test
  - b. Working Sub Directory → BankApp ( Project Root Directory under SVN)
  - c. Override 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:

The screenshot shows the Bamboo web interface for 'Build #1' of the 'Maven Bank Application'. The top navigation bar includes 'Bamboo', 'My Bamboo', 'Build', 'Deploy', 'Reports', and 'Create'. The main header shows 'Build projects / Bank Application / Maven Bank' and 'Build #1'. A green banner indicates 'Build #1 was successful – First build for this plan'. Below this, there are tabs for 'Summary', 'Tests', 'Commits', 'Artifacts', 'Logs', and 'Metadata'. The 'Summary' tab is active, displaying 'Build result summary' and 'Details'. The details section shows: 'Completed 13 Jan 2017, 3:51:06 PM – 37 seconds ago', 'Duration 8 seconds', 'Labels None', 'Agent Default Agent', 'Revision 2', and 'Total tests 11'.

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.

The screenshot shows the Bamboo web interface with the 'Logs' tab selected. The top navigation bar and header are the same as in the previous screenshot. The 'Logs' tab is highlighted with a red box. Below the tabs, the 'Logs' section is displayed. It shows a list of log entries for the 'Default Job Default Stage'. The logs include timestamps, build status, and test results. A 'Download or View' button is visible in the top right corner of the logs section, also highlighted with a red box. The bottom of the page features a warning icon and a message: 'Powered by a free Atlassian Bamboo evaluation license. Please consider purchasing it today.'

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

The screenshot shows the Atlassian Bamboo web interface. At the top, there's a navigation bar with 'Bamboo' and tabs for 'My Bamboo', 'Build', 'Deploy', 'Reports', and 'Create'. A search bar is on the right. Below the navigation bar, the main content area shows 'Build projects' for 'Bank Application'. A table lists the build details:

Plan	Build	Completed	Tests	Reason
Maven Bank	✓ #1	1 hour ago	11 passed	First build for this plan

At the bottom, there's a footer with the Atlassian logo and version information: 'Continuous integration powered by Atlassian Bamboo version 5.14.3.1 build 51416 - 05 Dec 16'.

## 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:

```
D:\vidavid\CI_For_Java\mastering-ci\tools\sonarqube-4.5.7\sonarqube-4.5.7\bin\wi
ndows-x86-64>StartSonar
wrapper      --> wrapper Started as Console
wrapper      Launching a JVM...
jvm 1       Wrapper (Version 3.2.3) http://wrapper.tanukisoftware.org
jvm 1       Copyright 1999-2006 Tanuki Software, Inc. All Rights Reserved.
jvm 1
jvm 1
jvm 1       WARNING - Unable to load the wrapper's native library 'wrapper.dll'.
jvm 1       The file is located on the path at the following location b
ut
jvm 1       could not be loaded:
jvm 1       D:\vidavid\CI_For_Java\mastering-ci\tools\sonarqube-4.5.7
\sonarqube-4.5.7\bin\windows-x86-64\.\lib\wrapper.dll
jvm 1       Please verify that the file is readable by the current user
```

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>

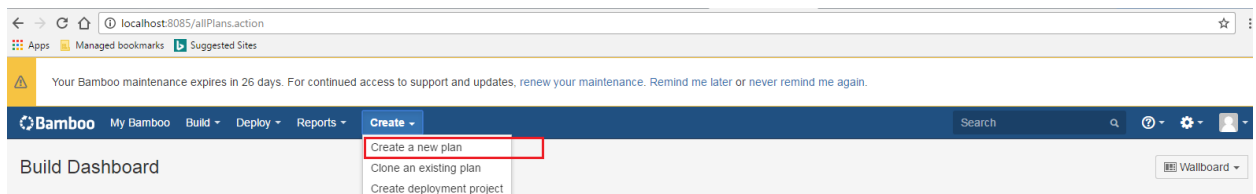
  <groupId>org.sonarsource.scanner.maven</groupId>

  <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

The project the new plan will be created in.

Project name\*

Project key\*

Eg. AT (for a project named Atlassian)

Plan name\*

Plan key\*

Eg. WEB (for a plan named Website)

Plan description

#### Link repository to new build plan

Repository host\* ☐ Previously linked repository

☒ Link new repository

Display name\*

#### Subversion details

Repository root URL\*

The location of the Subversion repository root (e.g. http://svn.collab.net/repos/svn)

Username

The subversion username (if any) required to access the repository

Authentication type

Password

The password required by the subversion username

Branch name

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)

✓ 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



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

Final tasks Are always executed even if a previous task fails

Drag tasks here to make them final

Add task

### Source Code Checkout configuration

[How to use the Source Code Checkout task](#)

Task description

Checkout Default Repository

☐ Disable this task

You can check out one or more repositories with this Task. You can choose to check out the Plan's *Default Repository* or specify a *Specific Repository*. You can add additional repositories to this Plan via the [Plan configuration](#).

Repository\*

Bank Repo

Default always points to Plans default repository.

Checkout Directory

BankApp

(Optional) Specify an alternative sub-directory to which the code will be checked out

☒ Force Clean Build
 

Removes the source directory and checks it out again prior to each build. This may significantly increase build times.

[Add repository](#)

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

## 6. Click Add Task button:

### 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

Final tasks Are always executed even if a previous task fails

Drag tasks here to make them final

Add task

No task selected

Select a task from the list on the left to configure it.

#### 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

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

Task types

All

Builder

Tests

Deployment

Source Control

Variables

variables

Injects Bamboo variables from a file with a simple "key=value"

Maven 1.x

Execute one or more Maven 1 goals as part of your build

Maven 2.x

Execute one or more Maven 2 goals as part of your build

Maven 3.x

Execute one or more Maven 3 goals as part of your build

Maven Dependencies Processor

Automatically update Plan dependencies by analysing the Maven pom file

Mocha Test Parser

Parse test results of Mocha executed with 'mocha-bamboo-reporter'

MSBuild

Run MSBuild as part of your build

Get more tasks on the Atlassian Marketplace or write your own

Cancel

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.

### Add new executable

This will be added to the local server capabilities.

Type **Maven 3.x**

Executable label

A label to uniquely identify this executable

Path

Please enter the Maven home value as your path. e.g.

**Add**

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.

### Add new JDK

This will be added to the local server capabilities.

JDK label

Java home

**Add**

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. Override 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:

The screenshot shows the Bamboo web interface for 'Build #1' of the 'Maven Bank Application'. The top navigation bar includes 'Bamboo', 'My Bamboo', 'Build', 'Deploy', 'Reports', and 'Create'. The main header shows 'Build projects / Bank Application / Maven Bank' and 'Build #1'. A green banner at the top states 'Build #1 was successful – First build for this plan'. Below this, there are tabs for 'Summary', 'Tests', 'Commits', 'Artifacts', 'Logs', and 'Metadata'. The 'Summary' tab is active, displaying 'Build result summary' and 'Details'. The details section shows: 'Completed 13 Jan 2017, 3:51:06 PM – 37 seconds ago', 'Duration 8 seconds', 'Labels None', 'Agent Default Agent', 'Revision 2', and 'Total tests 11'.

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.

The screenshot shows the Bamboo web interface with the 'Logs' tab selected. The top navigation bar and header are the same as in the previous screenshot. The 'Logs' tab is highlighted with a red box. Below the tabs, the 'Logs' section is displayed. It shows a list of logs generated by the jobs in this plan. The first log is expanded, showing the following text: '13-Jan-2017 18:51:06 \$-->1', '13-Jan-2017 18:51:06 \$-->1', '13-Jan-2017 18:51:06 \$-->0', '13-Jan-2017 18:51:06 \$-->3', '13-Jan-2017 18:51:06 \$-->6', '13-Jan-2017 18:51:06 \$-->7', '13-Jan-2017 18:51:06 \$-->1', '13-Jan-2017 18:51:06 \$-->1', '13-Jan-2017 18:51:06 \$-->3', '13-Jan-2017 18:51:06 \$-->6', '13-Jan-2017 18:51:06 \$-->7', '13-Jan-2017 18:51:06 Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.068 sec', '13-Jan-2017 18:51:06 Results:', '13-Jan-2017 18:51:06 Tests run: 11, Failures: 0, Errors: 0, Skipped: 1', '13-Jan-2017 18:51:06 [INFO] -----', '13-Jan-2017 18:51:06 [INFO] BUILD SUCCESS', '13-Jan-2017 18:51:06 [INFO] -----', '13-Jan-2017 18:51:06 [INFO] Total time: 4.636s', '13-Jan-2017 18:51:06 [INFO] Finished at: Fri Jan 13 18:51:06 IST 2017', '13-Jan-2017 18:51:06 [INFO] Final Memory: 18M/16M'. A red box highlights the 'Download or View' link in the top right corner of the logs section. At the bottom, there is a warning icon and a message: 'Powered by a free Atlassian Bamboo evaluation license. Please consider purchasing it today.'

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

The screenshot shows the Bamboo web interface. At the top, there's a navigation bar with 'Bamboo', 'My Bamboo', 'Build', 'Deploy', 'Reports', and 'Create'. Below this, the project name 'Bank Application' is displayed. A table shows build details:

Plan	Build	Completed	Tests	Reason
Maven Bank	✓ #3	46 seconds ago	11 passed	Manual run by David Vijayalakshmi

At the bottom, it mentions 'Continuous integration powered by Atlassian Bamboo version 5.14.3.1 build 51416 - 05 Dec 16' and provides links for 'Report a problem', 'Request a feature', 'Contact Atlassian', and 'Contact Administrators'.

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>

The screenshot shows the SonarQube dashboard for 'BankApp'. It displays various quality metrics:

- Lines Of Code:** 189
- Files:** 10
- Functions:** 22
- Java Directories:** 6
- Lines:** 282
- Classes:** 10
- Statements:** 68
- Accessors:** 12
- Duplications:** 0.0%
- Complexity:** 1.8 /function, 3.9 /class, 3.9 /file
- Total:** 39
- SQL Rating:** A
- Technical Debt Ratio:** 4.3%
- Technical Debt:** 4h 3min
- Issues:** 30
- Issues Breakdown:** 0 Blocker, 0 Critical, 15 Major, 13 Minor, 2 Info
- Directory Tangle Index:** 0.0%
- Dependencies To Cut:** 0
- Unit Tests Coverage:** 100.0%
- Unit Test Success:** 100.0%
- Failures:** 0
- Errors:** 0
- Tests:** 10
- Skipped:** 1
- Execution Time:** 138 ms

### **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! 😊**