Integrating Bamboo with Checkmarx

**Summary**

The purpose of this document is to explain how to run a Checkmarx scan from an existing Bamboo build plan. Checkmarx will scan the source code that is in the build working directory folder and return a result status to Bamboo.

The Checkmarx scan is configured as a Bamboo build step that executes a windows batch script. The script used, Checkmarx\_scan\_Bamboo.bat (provided by Checkmarx), will zip the source code that is in the project’s workspace and will send it to the Checkmarx server for scanning. At the end of the scan Checkmarx will create a pdf report of the scan and return a status of Success or Failure based on the number of vulnerabilities that were found in the scan. The return status will be “Failed” if the number of high risk vulnerabilities that were found in the scan is greater than the “High Risk Threshold” that is defined in the script (Checkmarx\_scan\_Jenkins.bat) OR if the number of medium risk vulnerabilities that were found in the scan is greater than the “Medium Risk Threshold” that is defined in the script.

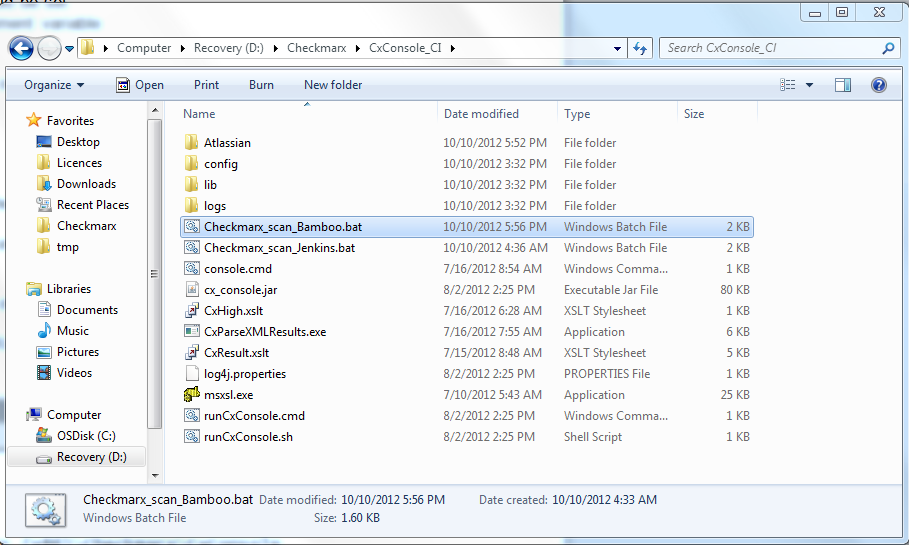
Prerequisite:

* Java must be installed on the CI-Server ,
* JAVA\_HOME environment variable should be set
* Java.exe should be in the PATH environment variable

Note: By default, Bamboo’s build plan name is used to create the Checkmarx project. Therefore the Bamboo plan name must not contain spaces or special characters. This can be changed by modifying the definition of JOB\_NAME in the Checkmarx\_Scan\_Bamboo.bat

The steps to configure a scan from Bamboo are the following:

1. Decompress the attached file CxConsole\_CI.zip on the Continuous Integration Server. The contents of the folder should be:



1. Edit file Checkmarx\_Scan\_Bamboo.bat and set the following parameters:

CX\_CONSOLE\_PATH is the location when you unzipped the CxConsole package

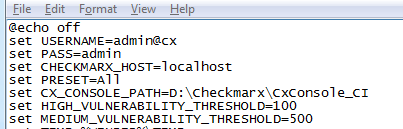
USERNAME is your Checkmarx user account

PASS is your Checkmarx password

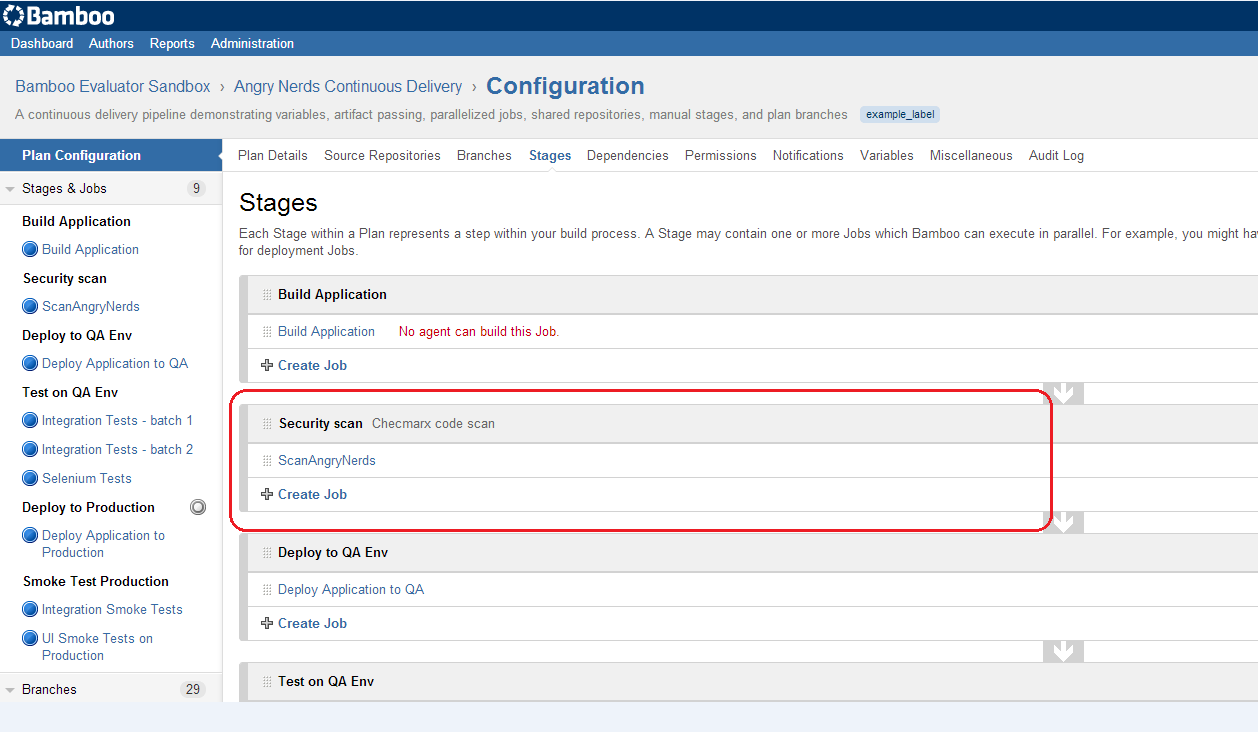
CHECKMARX\_HOST is the name or ip address of the Checkmarx Server

PRESET is the Checkmarx preset of the scan

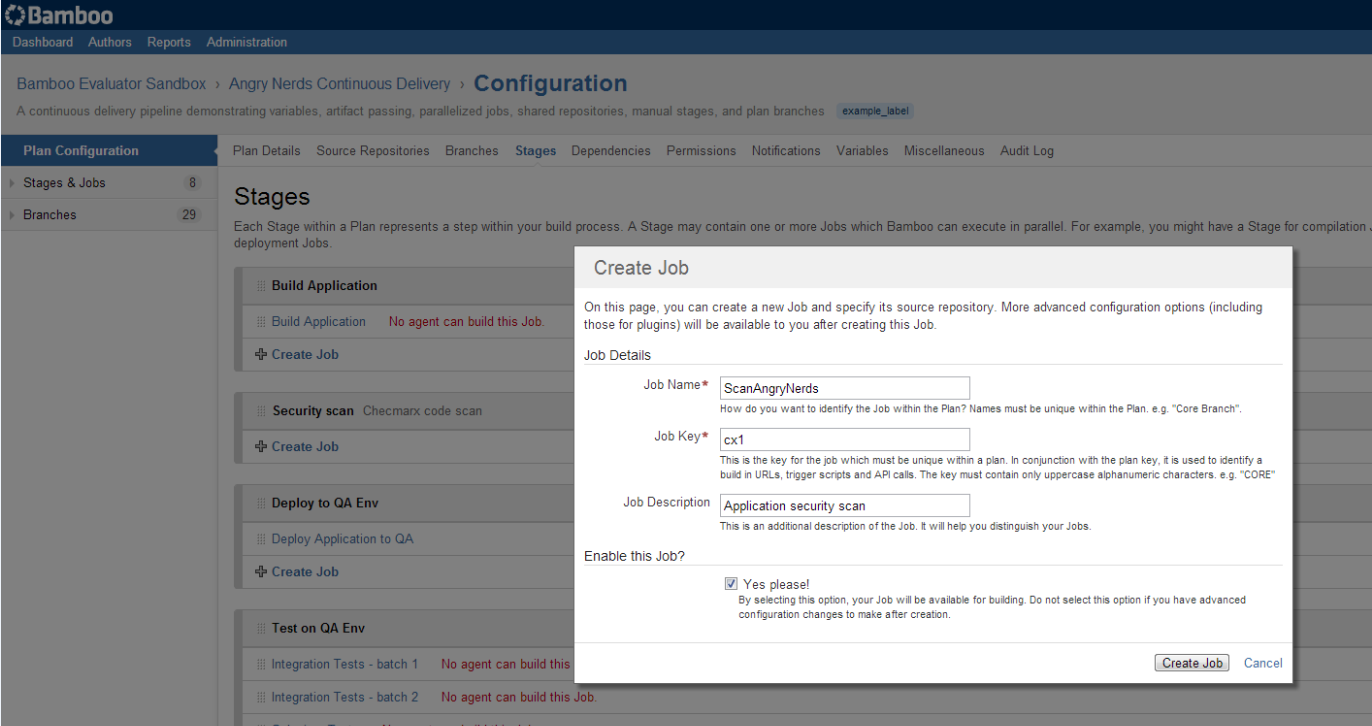
JOB\_NAME is the name of the Checkmarx project. Please note that CxServer\SP\Company\USERS\%JOB\_NAME% should resemble the full name of your project. If you have defined companies or teams in your Checkmarx system, please edit them accordingly in the script file.



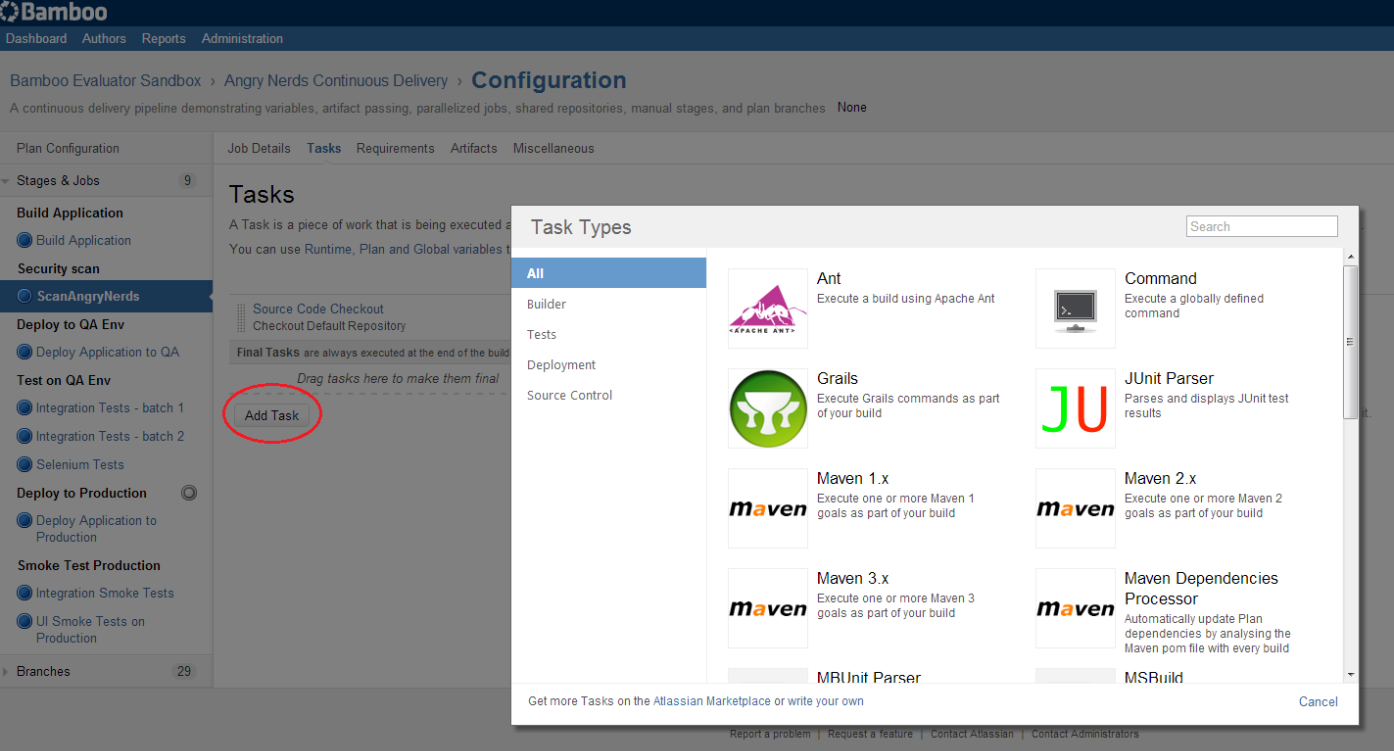
1. Define a dedicated Stage and a Job in the stage that the will launch the Checkmarx\_scan\_Bamboo.bat batch file. For example, create a “Security scan” stage:



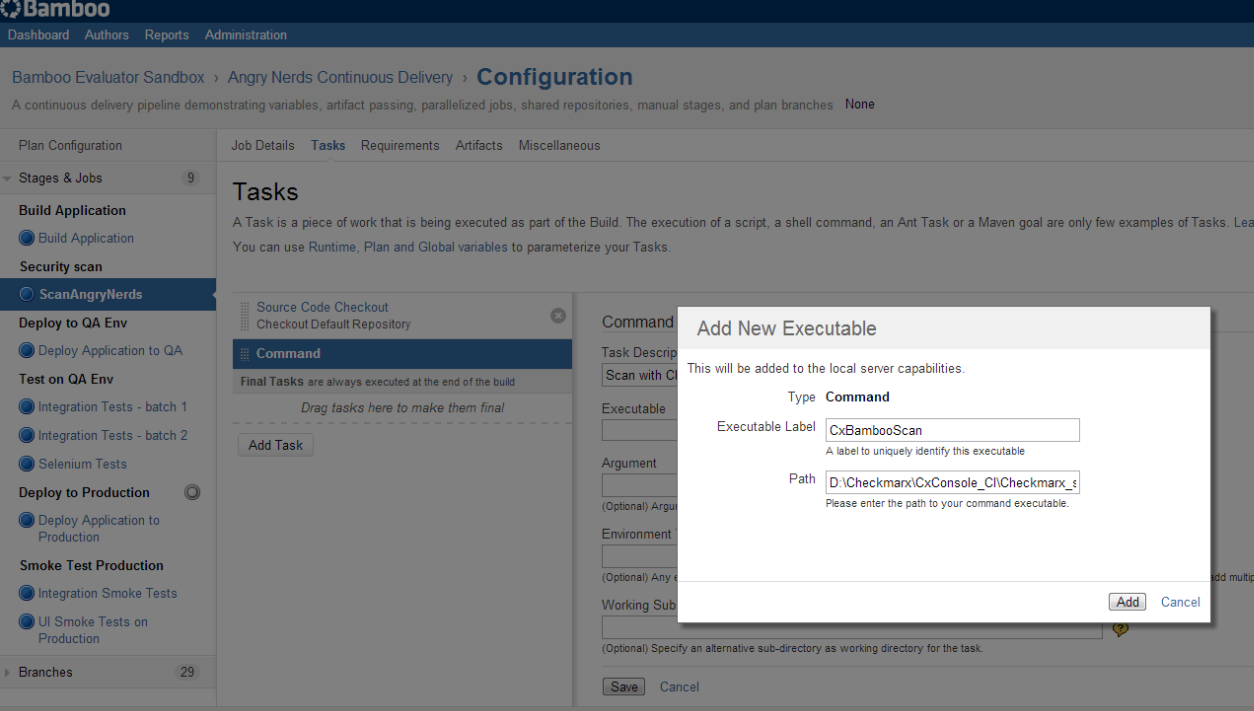
And in it, create a Job:



1. On the Job screen, choose the “Tasks” tab and press “Add task” to create the task that will launch the batch file, choose “Command” type.



5) Press the “Add new Executable” button, and in the below screen, enter the full path to the batch file. In our example: D:\Checkmarx\CxConsole\_CI\Checkmarx\_scan\_Bamboo.bat

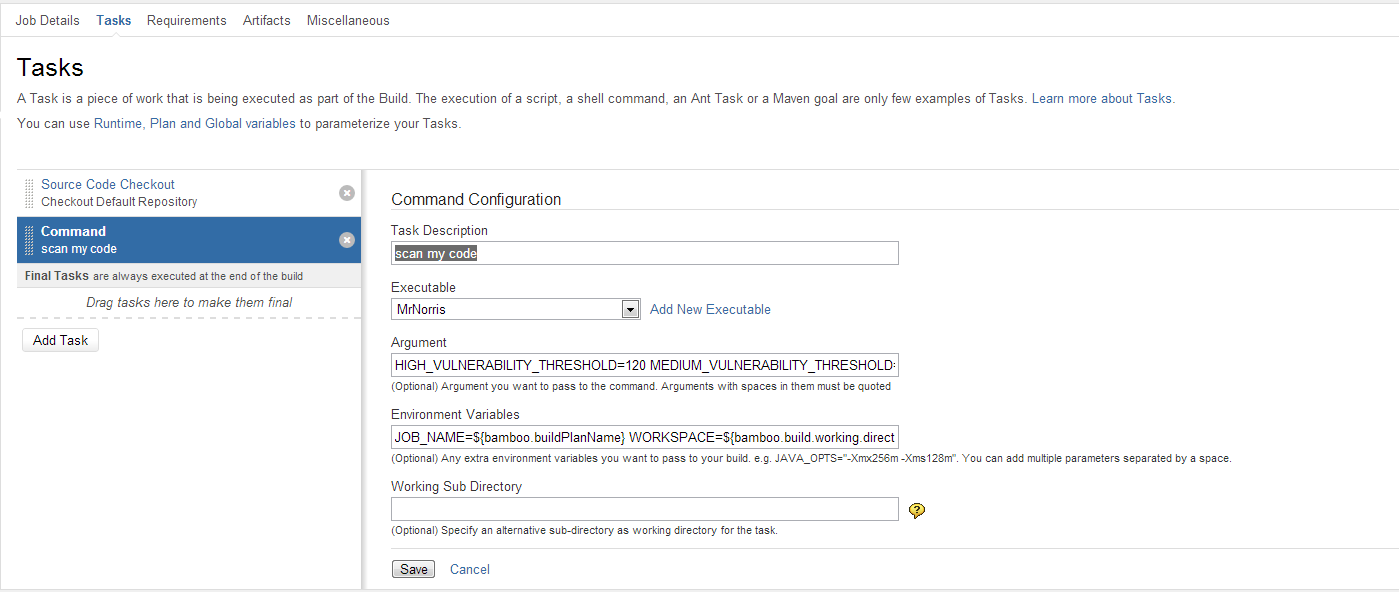


6) In the below screen, write the following in the “Arguments” (edit the values as see fit):

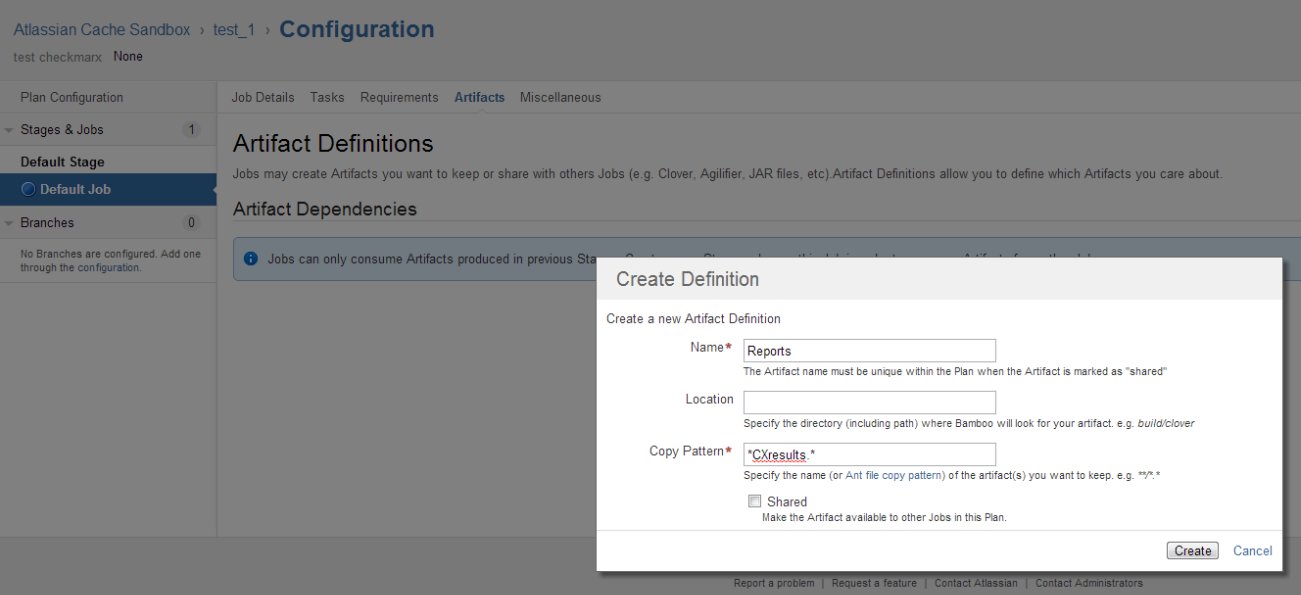
HIGH\_VULNERABILITY\_THRESHOLD=120 MEDIUM\_VULNERABILITY\_THRESHOLD=200

“Environment Variables”:

JOB\_NAME=${bamboo.buildPlanName} WORKSPACE=${bamboo.build.working.directory}



7) On the new created Job screen, choose the “Artifacts” tab and press “Create definition” and enter the below values. Name: CxReports Copy Pattern: \*CXresults.\*

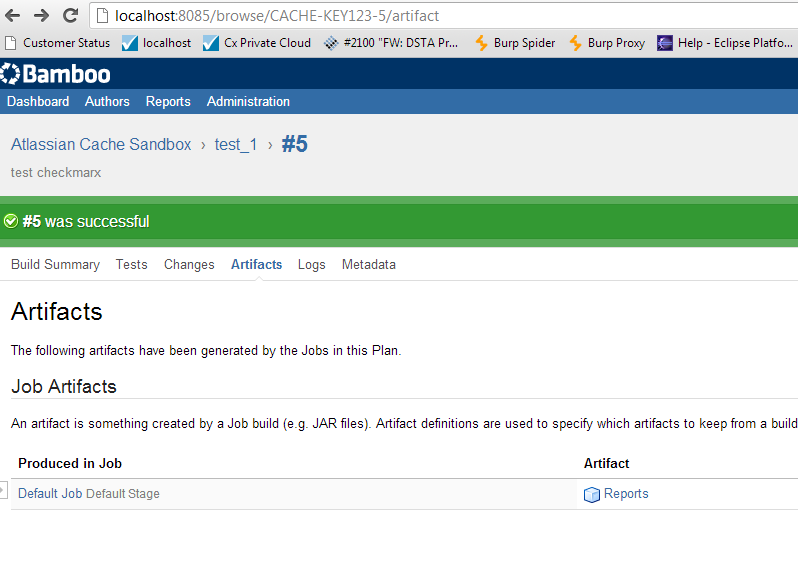


8) That’s it! You’re now ready to run the project.

If the scan detects more than HIGH\_VULNERABILITY\_THRESHOLD high severity vulnerabilities or more than MEDIUM\_VULNERABILITY\_THRESHOLD medium severity vulnerabilities, the stage will fail and the next stage of the project will not be executed.

Otherwise, the stage will complete and the next stage will begin.

The results of the scan will be accessible from the “Artifacts” section.



And should look something like this: