

Bamboo Pipeline Setup

Build plan	2
Deployment setup:	3
Setting of pipeline to push code from DEV to QA.....	8
Figure 1: Agent installation	2
Figure 2: Build config	3
Figure 3: Deployment project	3
Figure 4: Configuration of deployment project	4
Figure 5: Configuration of Deployment stages	5
Figure 6: SCP details.....	6
Figure 7: Trigger setup	7
Figure 8: Showing pipeline.....	8
Figure 9: Showing two environment setup.....	8
Figure 10: Functional verification step	9
Figure 11: QA environment setup.....	9
Figure 12: QA environment trigger.....	10


Build plan

a. Run agent by downloading and running “java -jar /home/ec2-user/atlassian-bamboo-agent-installer-5.13.2.jar <http://ip:8085/agentServer/>”

Installing a remote agent

To install a remote agent, please follow these instructions:

1. Ensure that you have Java Runtime Environment 8.0 or later installed on the agent machine.
2. Download the remote agent JAR file to a directory on the agent machine.



Running a remote agent

Once installed, you can run the remote agent by executing the following command line from the directory containing the remote agent jar file:

```
java -jar atlassian-bamboo-agent-installer-5.13.2.jar http://52.53.228.239:8085/agentServer/
```

This will start a service wrapper for your agent, which will automatically restart in case of failure. You may also add extra system properties like `-Dbamboo.home=...` to customise the agent's home location. For more information, see our [Bamboo remote agent installation guide](#).

Running the agents without the service wrapper

For customers wanting to run the Bamboo agent without the service wrapper, the direct agent jar is available at [bamboo-agent-5.13.2.jar](#). For more information about using the JAR directly please consult [our documentation](#).

Figure 1: Agent installation

b. Configure maven in a bin/configuration (create `bamboo-capabilities.properties` file) file under `/root/bamboo-agent-home/bin`.
<https://confluence.atlassian.com/bamboo/configuring-remote-agent-capabilities-using-bamboo-capabilities-properties-289276849.html>

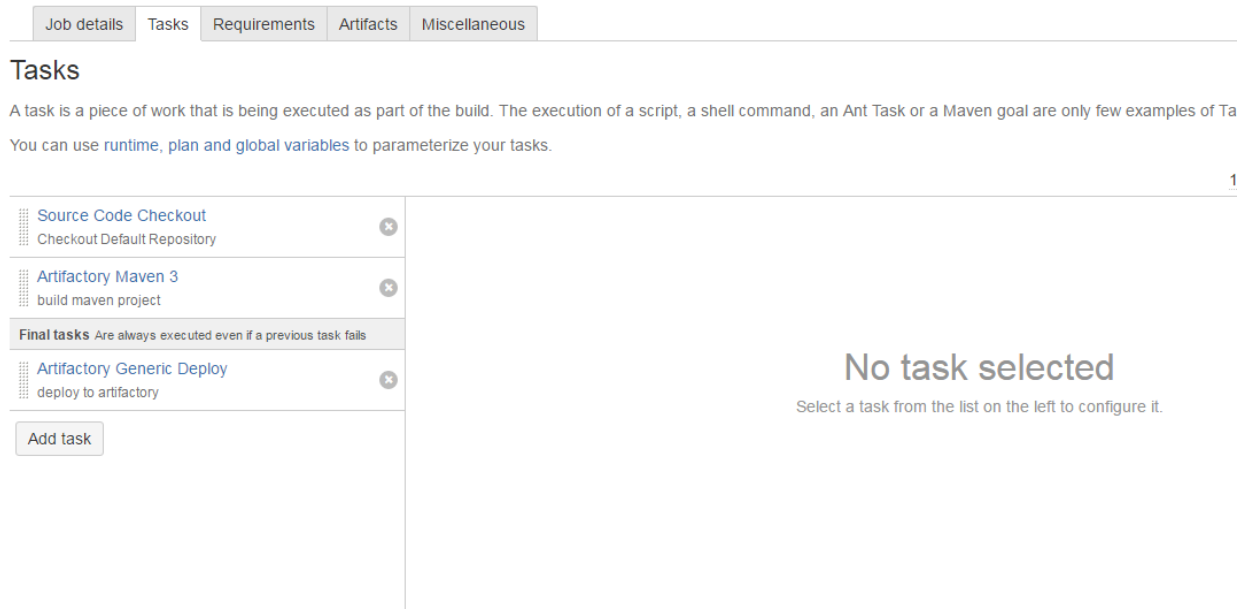


Figure 2: Build config

Deployment setup:

a: Under create/create deployment project.



Figure 3: Deployment project

b. Link build plan to Deployment project:

Update deployment project

[How deployments work](#)

A deployment project defines which build plan you get your artifacts from, and contains the environments you want to deploy to.

Deployment project details

Name *

ARTI

Description

Link to build plan

[How deployment releases work](#)

Shared artifacts of the selected plan will be bundled into releases. Releases will be deployed to the environments.

Build plan *

Artifactory_Demo > Artifactory_Demo_PLAN

Start typing the plan name or use the down arrow to select a plan. The selected plan will be used as the source for artifacts that will be deployed as a release.

☒ Use the main plan branch

Currently

master

☐ Use a custom plan branch

Save deployment project

Cancel

Figure 4: Configuration of deployment project

c. Name environment as DEV/QA/PROD.

d. Declare stages:

Update tasks: DEV

What tasks need to happen to make this deployment a success

Clean working directory task

Artifact download

Download release contents

SCP Task

Release

Final tasks Are always executed even if a previous task fails

Drag tasks here to make them final

Add task

1 agent has the capabilities to deploy this environment

No task selected

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

Back to deployment project

Continuous integration powered by Atlassian Bamboo version 5.13.2 build 51316 - 28 Sep 16

Figure 5: Configuration of Deployment stages

e. SCP task copies artifacts to Mule server:

Update tasks: DEV View deployment summary

What tasks need to happen to make this deployment a success

1 agent has the [capabilities](#) to deploy this environment

Clean working directory task

Artifact download
Download release contents

SCP Task
Release

Final tasks
Are always executed even if a previous task fails

Drag tasks here to make them final

Add task

SCP Task configuration

[How to use the SCP task](#)

Task description

☐ Disable this task

Host*

Hostname or IP address of the remote host

Username*

Username you want to use to access the remote host

Authentication Type*

Key without passphrase

☐ Change SSH Key

Artifact

ART-ARP: zip

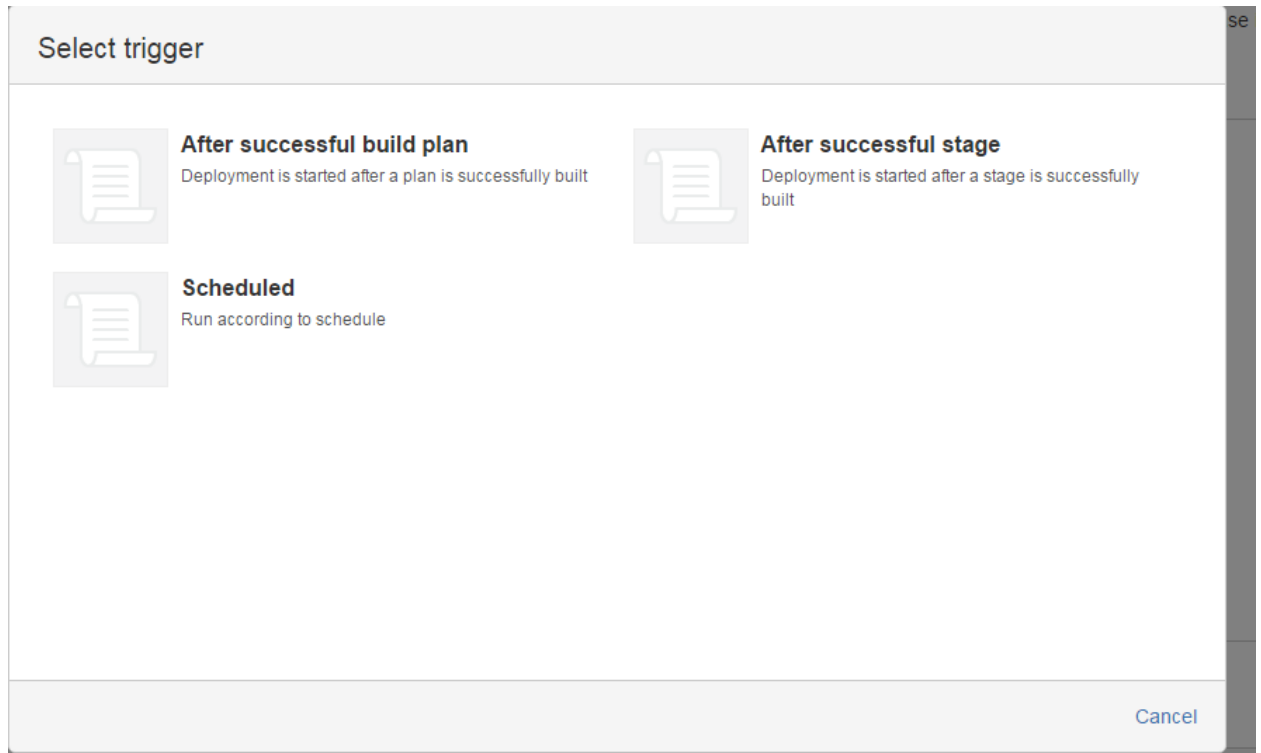
Remote Path

Advanced Options

Save Cancel

Figure 6: SCP details

f. Configure trigger for deployment:



The screenshot shows a dialog box titled "Select trigger". It contains three options, each with a document icon and a description:

- After successful build plan**
Deployment is started after a plan is successfully built
- After successful stage**
Deployment is started after a stage is successfully built
- Scheduled**
Run according to schedule

A "Cancel" button is located in the bottom right corner of the dialog box.

Figure 7: Trigger setup

g. Run deployment.

Setting of pipeline to push code from DEV to QA:

a. Logical trigger point to deploy DEV code to QA environment would be “verification of functional test results deployed in DEV”. We have tried to achieve that using bamboo trigger events and components. Snippet below shows script which checks “http:status code only if it’s values comes as ‘200’ DEV deployment stages would be marked completed else scripts throws error”.

The screenshot shows the Bamboo interface for Build #13. The top navigation bar includes links for My Bamboo, Build, Deploy, Reports, and Create. The build status is indicated by a green bar with the text "#13 was successful – Manual run by POC". Below this, the build result summary is displayed, including details such as completion time (25 Oct 2016, 8:32:13 AM), duration (8 minutes), agent (localhost (3)), and revisions (demo-repor3, channelDemo). A yellow highlight is placed on the green status bar. A red arrow points from the 'DEV' environment in the deployment project table to the 'DEV' environment in the build details table.

Environment	Status	Actions
DEV	SUCCESS	
QA	SUCCESS	

Figure 8: Showing pipeline

The screenshot shows the Bamboo Configuration page for the Demo project. The configuration includes the source build plan, available artifacts, and a list of environments. The environments are DEV and QA, both with a status of SUCCESS. The configuration also includes buttons for Edit build plan, Release versioning, Project permissions, and Add environment.

Environment	Status	Actions
Environment: DEV	SUCCESS	Deploy Edit...
Environment: QA	SUCCESS	Deploy Edit...

Figure 9: Showing two environment setup

What tasks need to happen to make this deployment a success

2 agents have the capabilities to deploy this environment

Clean working directory task

Artifact download

Download release contents

SCP Task

Script

Final tasks Are always executed even if a previous task fails

Drag tasks here to make them final

Add task

Script configuration

Task description

☐ Disable this task

Interpreter

/bin/sh or cmd.exe

Run your script with /bin/sh or cmd.exe

Script location

Inline

Script body*

```
1 res=$(curl -X GET -s -o /dev/null -I -w "%{http_code}" "http://52.53
2 echo "result is: $res"
3 if [ $res -eq 200 ];
4 then
5     echo "result is $res on request"
6 else
7     cd 123
8 fi
```

Figure 10: Functional verification step

Environment: QA

Deploy

Actions

Minimise

How you want to deploy

Tasks define the steps involved in deploying a release to the environment.

Edit tasks

Other environment settings

These settings are not strictly necessary for your deployment to run, but they can be very helpful and allow you to make Bamboo deployments go just right.

Triggers 1

Agents assignment

Notifications 0

Variables 0

Environment permissions

+ Add environment

Figure 11: QA environment setup

Edit triggers: QA [How deployment triggers work](#)

Set triggers to specify how and when the deployment will be triggered automatically. When a deployment is automatically triggered, a new release is created from the latest successful build result of the linked plan.
If you want to deploy to this environment manually, use the "Deploy" action at any time.

After successful deployment
[Add trigger](#)

Trigger configuration

Trigger description

☐ Disable this trigger

Triggering environment*

DEV

[Save trigger](#) [Cancel](#)

Figure 12: QA environment trigger