Chapter 3

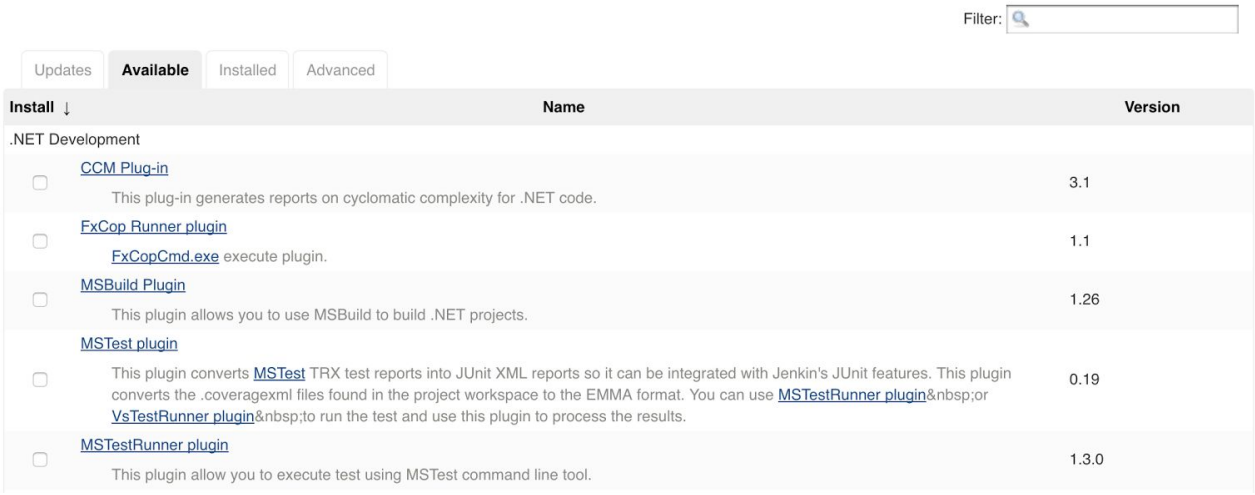
Lab 3.5 - Integrating Jenkins and Artifactory

Prerequisites:

1. Running the Artifactory instance with port forwarding to 8081 (Chapter 3, Lab 3.4).
2. Running the Jenkins instance with port forwarding to 8080 (Chapter 2).
3. Working the build job using a forked “spring-petclinic” (Chapter 3, Lab 3.3).

Step 1: Install the Artifactory Plugin

1. From the Jenkins main page, navigate to the “Plugin Manager”:
“Manage Jenkins” > “Plugin Manager”.



2. Click on the “Available” tab.

3. Search for “Artifactory”.

4. Check the box next to “Artifactory Plugin”.

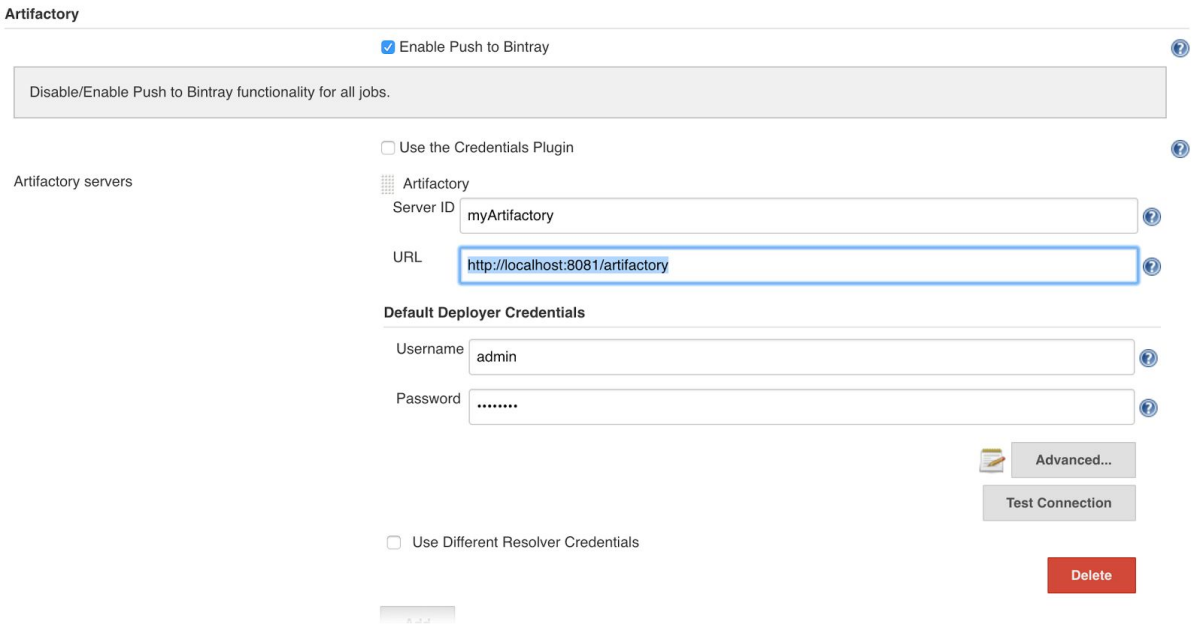
5. At the bottom of the screen, several buttons will appear. Click on the button that says: “Download now and install after restart”.

6. This opens a new page showing the plugin being installed. On this page, check the box next to “Restart Jenkins when installation is complete and no jobs are running”. Once the download is complete, Jenkins will restart with the new plugin installed.

Step 2: Connect Artifactory to Jenkins

1. From the Jenkins main page, navigate to the system configuration menu “Manage Jenkins” -> “Configure System”.

2. Scroll down to the “Artifactory” section.



3. Fill out the “Server ID” section with an appropriate name. This name will be used later to identify the Artifactory server.

4. Fill out the “URL” section with: **`http://localhost:8081/artifactory`**.

5. You can access the destination Artifactory repositories under “Default Deployer Credentials”. Fill out the user information using the appropriate login credentials: username: ”**`admin`**”, password: “**`<whatever was set in the previous lab>`**”.

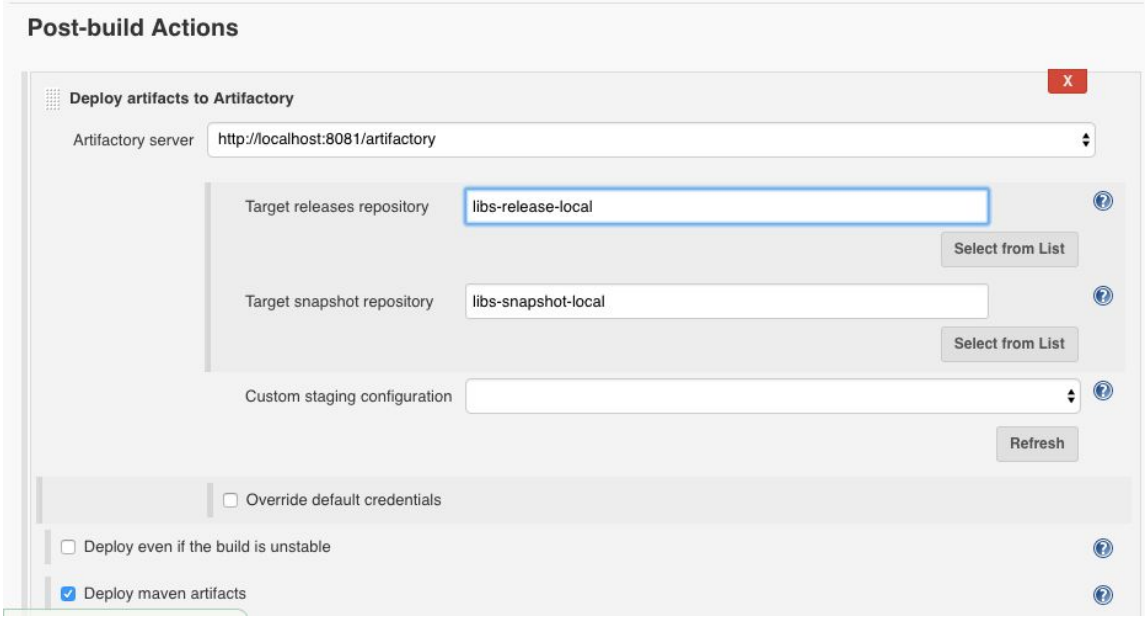
6. Click on the “Test Connection” button to ensure that the Artifactory server can be reached. If the connection test fails, double check the URL to make sure that it matches the one on the destination artifact repository.

Step 3: Transform the Build Job Into a Deploy Job

1. Navigate to the configuration page of the existing “spring-petclinic” build job (if it does not exist, please refer to Chapter 3, Lab 3.3).

2. Under “Post-build Actions”, select the “Run Only if Build Succeeds” bubble. Next, choose “Deploy artifacts to Artifactory” from the “Add Post-build Action” dropdown menu.

3. Fill out the sections with the appropriate information. In order to edit the “Target releases repository” and “Target snapshot repository” sections, press the “Different Value” button, and type the target repository in (please refer to the image below).



4. Save and exit the configuration page.

Step 4: Trigger the Deploy Job

1. Edit the **`pom.xml`** file in the cloned repository. Change the version tag to have the current snapshot **`version + .1`** (should read **`1.0.1`**).

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="ht
    <modelVersion>4.0.0</modelVersion>
    <groupId>org.springframework.samples</groupId>
    <artifactId>spring-petclinic</artifactId>
    <version>1.0.0-SNAPSHOT</version>

```


2. Now, commit and push the altered code:

```
$ git add .
$ git commit -m 'Updated version notice in pom.xml'
$ git push origin master
```

3. The build should be pushed from GitHub to Jenkins and from Jenkins into Artifactory. You can confirm it once the build is complete, by logging into Artifactory and looking at the artifacts in the Artifact Repository Browser.

4. For more information on Artifactory, please refer to the following resources: Artifactory 4.0 Jenkins Integration and Sample projects for training and testing CI setup with Artifactory.

