Deploying Snapshots to oss.jfrog.org

Overview

What is OJO

oss.jfrog.org, or **OJO** for short, is an Artifactory Cloud instance for hosting your maven-compatible build snapshots, provided free of charge for selected opensource software projects.

All projects in OJO are public (i.e., all the artifacts and builds can be viewed by anyone).

Existing Bintray users are granted deploy permissions to relevant folders in Artifactory, according to the Maven Group ID of the packages they build.

Target Audience

This page is designed to help OSS contributors who want a free repository to host build snapshots, and eventually publish release versions to distribution via Bintray.

At a Glance

The process of on-boarding to OJO, working with it to deploy continuous snapshots, and finally, promoting these snapshots from OJO to Bintray for distribution involves three simple steps:

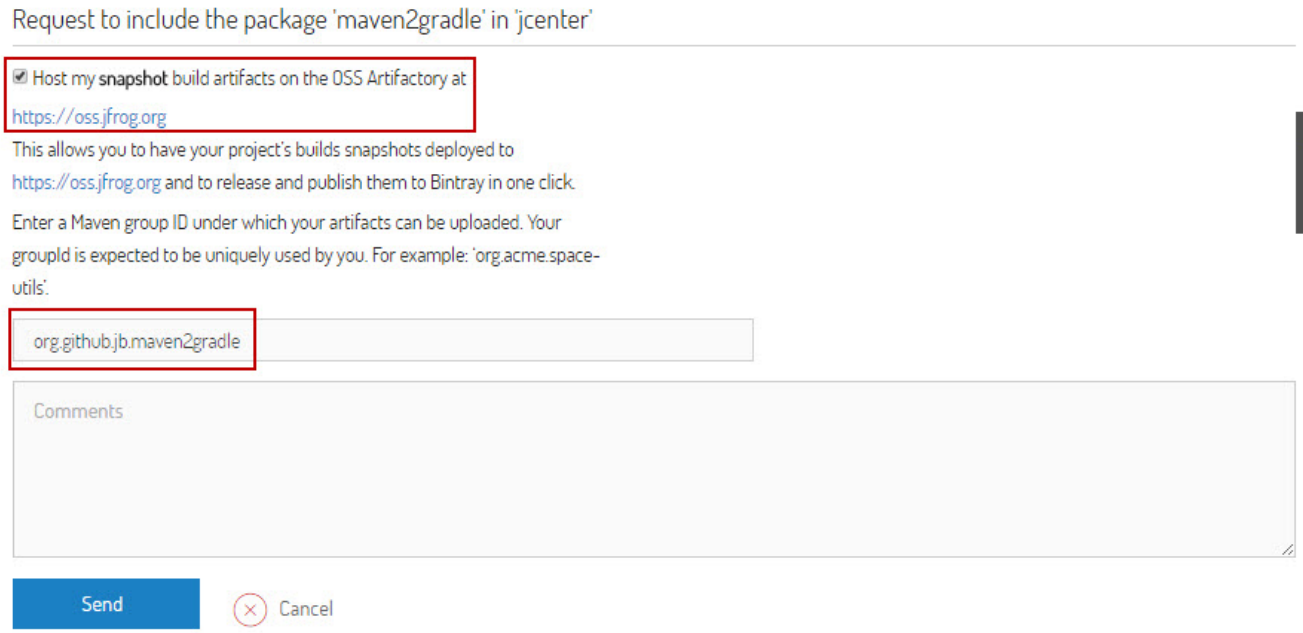
1. Creating an account on OJO
2. Building and deploying to the OJO Artifactory
3. Promoting a snapshot build to Bintray

Page Contents

Getting Started with OJO

To get account on OJO you must first have an account on Bintray.

1. Create a Maven repo on Bintray if it does not exist yet, and create your package inside this repo.
   You can give your package any logical name, for example: `maven2gradle`
2. Ask for inclusion of the package in JCenter, by clicking the "**Add to JCenter**" button in the package main page.
   In the request form, check "**Host my snapshot build artifacts on the OSS Artifactory at https://oss.jfrog.org**" and enter the desired Maven group ID for your package.
   For example: `org.github.jbaruch.maven2gradle`



After your request has been approved by the Bintray Team (usually within a few hours), you'll receive a confirmation email on the inclusion of your package in JCenter and the creation of your new OJO account.

OJO is Artifactory!

OJO is just a regular Artifactory Pro server, so getting familiar with Artifactory is recommended.

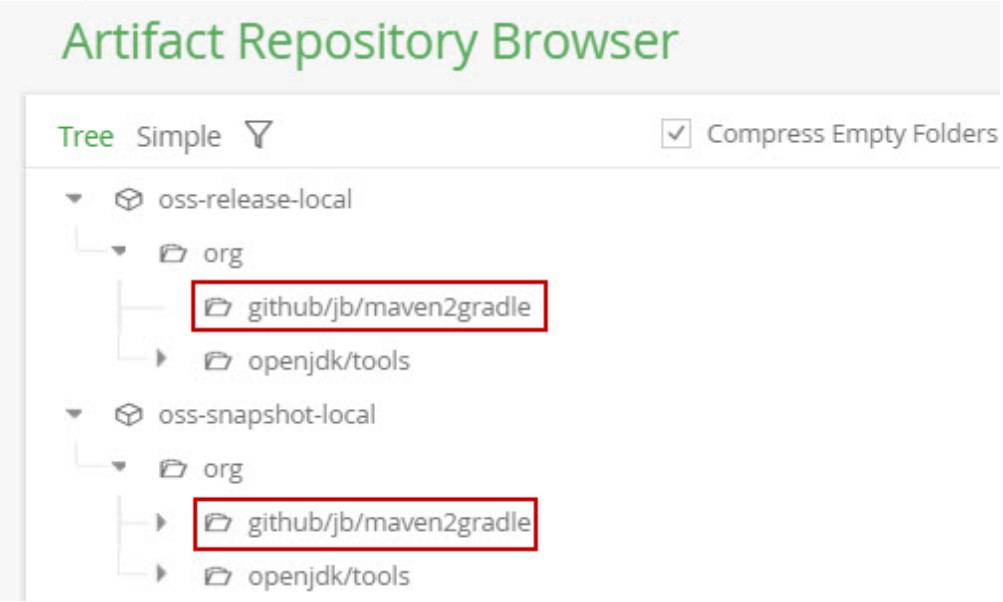
Bintray Organizations Support

When requesting an OJO account for a repository belonging to an Bintray organization, the permissions in OJO will be granted to all the organization members, not only the member who asked for the OJO account.

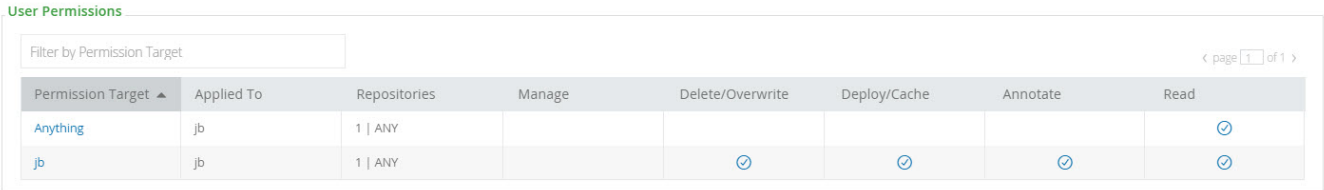
Working with OJO

Once your OJO account has been created, you (and all the team members in the case of an organization) should be able to login to OJO using your **Bintray username** and **API key** as the password.

You will see that a folder corresponding to the Maven Group ID has been created in OJO in the `oss-release-local` and the `oss-snapshot-local` repositories:



You have deploy permissions to these folders:



Resolving from and Publishing to OJO

There is nothing unusual about working with repositories in OJO. You can configure your build tool to resolve release and snapshot dependencies from the `libs-release` and the `libs-snapshot` OJO virtual repositories, respectively; and to deploy build snapshots to the `oss-snapshot-local` repository. As long as the `<groupId>` in your pom (for Maven) or the `project.group` (for Gradle) matches the group ID you requested during onboarding, the deployment should succeed.

Please consult the Artifactory documentation on how to set up Maven or Gradle for resolution and deployment.

Artifactory Build Info is a must!

In order to release and promote snapshots to Bintray you need to deploy a Build Info BOM to Artifactory. The easiest way to achieve this automatically it is to use the Build Integration feature of Artifactory or the Gradle Artifactory Plugin; These and other options are described in the next section.

Releasing to Bintray

Currently, you have two ways to deploy artifacts to Bintray:

1. **Promoting a Release Build**
   This will promote snapshot artifacts to release and then deploy them to Bintray:
   a. Use promotion from the Jenkins Artifactory plugin - This allows you to use the Jenkins UI to promote the snapshot artifacts from a selected job run.
   b. Invoking promotion with REST - This allows promotion of a build created with any build server/tool and full programatic automation of the promotion process.
2. **Uploading Release Artifacts**
   Directly upload deployed release artifact to Bintray. If you have a released version of an artifact or a build, you can deploy them to Bintray using the regular Bintray integration.

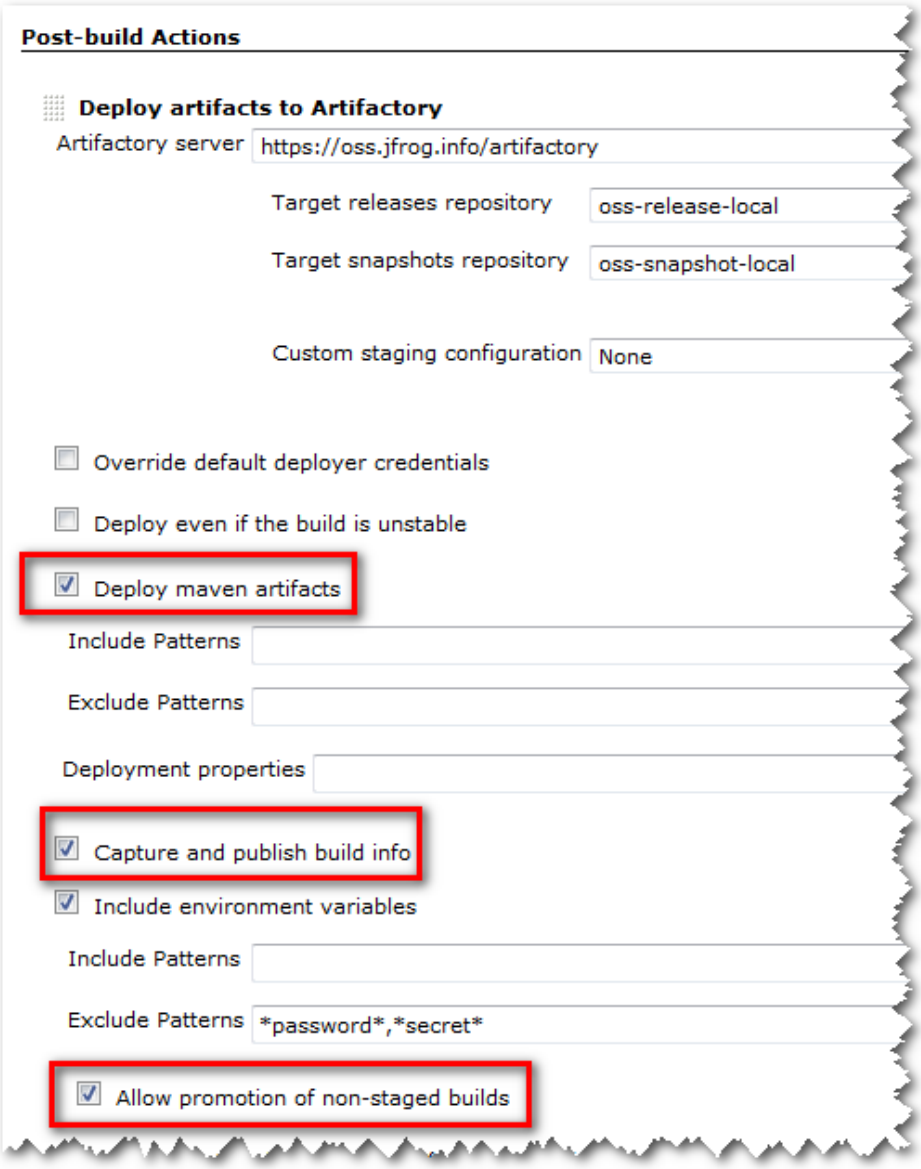
Promoting a Release Build

Promoting a Build from Jenkins

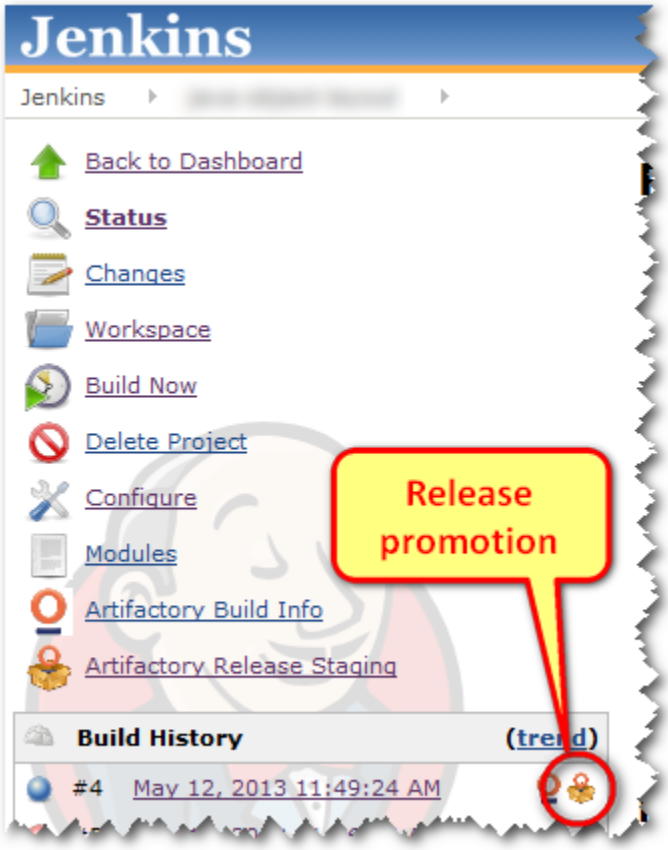
Promotion from Jenkins is performed by invoking a custom "**snapshotsToBintray**" promotion plugin. Here's what you need to do:

1. Install the Jenkins Artifactory Plugin and configure Artifactory servers and repositories as described in the Jenkins documentation. You should configure the `oss-release-local` and `oss-snapshot-local` as release and snapshot targets, respectively.

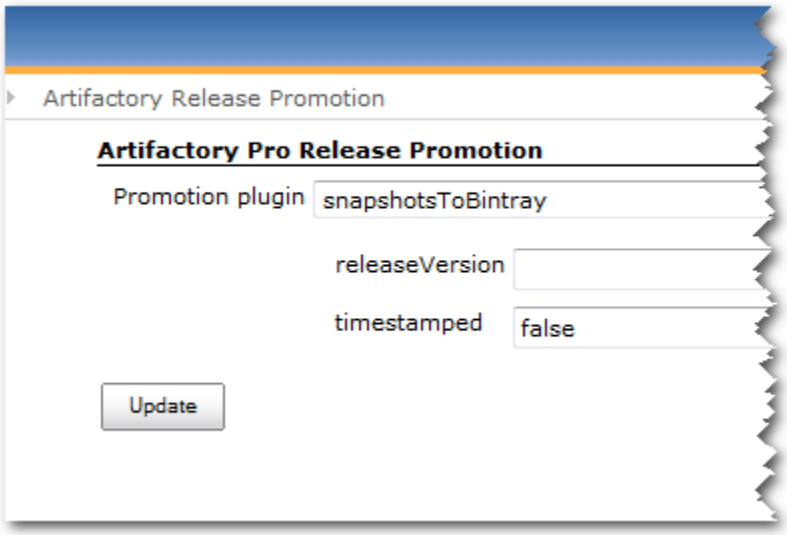
2. In your build configuration, add the "Deploy artifacts to Artifactory" post-build action and check "Deploy Maven artifacts", "Capture and publish build info" and "Allow promotion of non-staged builds":



3. Run your build. Upon successful completion, the build result page will have a link to the "Artifactory Release Promotion" action:



4. In the promotion configuration screen, select "snapshotsToBintray" promotion plugin:



There are two parameters to configure here:

   a. Override the release version. By default, the version is calculated by removing the -SNAPSHOT suffix from the snapshot version, e.g. `1.0-SNAPSHOT` will be released to Bintray as version `1.0`.
   Specifying a value in this field overrides the default version scheme.
   b. Append a timestamp to the version. This will add a timestamp string (in Maven's timestamp format: `yyyyMMdd.HHmmss`) to the release version. Values of `true`, `y` or `1` will cause the timestamp suffix to be appended.

5. Click the "Update" button. Your release artifacts are now uploaded to Bintray.

Promoting a Build Using REST API

If you don't use Jenkins or if you need fully automated promotion, you can issue an HTTP PUT request that will trigger promotion and release to Bintray. Promotion still operates on a Build Info BOM, previously saved in Artifactory.
Here's what you need to do:

1. Deploy a build to Artifactory in one of the following ways:

    a. Using a build server with an Artifactory plugin. Plugins currently exist for Jenkins, Hudson, Bamboo and TeamCity. Please see the Artifactory Build Integration documentation for further instructions on getting the build info BOM into Artifactory.
    b. Using the Gradle Artifactory Plugin.
    c. Configure Maven to use Artifactory Listener as described here.
2. Execute the build promotion plugin call. The call accepts the same parameters as the invocation of Jenkins promotion plugin. Here's an example using CURL:

```
curl -X POST -u bintrayUser:apiKey http://oss.jfrog.org/api/plugins/build/promote/snapshotsToBintray
/buildName/3
```