(DEPRECATED) Releasing jclouds the old and painful way

Checklist Summary of Steps Involved in Releasing

1. Email the dev list to let them know the release process is beginning and ask all to clean up any JIRA issues they are familiar with.
2. Have a GPG signing key configured.
3. Clone all repos to be released, if you haven't already.
4. Check out the branch to be released in each repo.
5. Update any POM properties in the repos pointing at the SNAPSHOT version and change those to the release version.
6. Run mvn release commands for each repo, in order, as described below. If building a 1.6.x release, use a Java 6 JDK
7. Push the release commits and RC tags to the remote repos.
8. Close the Nexus staging repository.
9. Copy the source release tarballs, signatures and hashes to a new release candidate directory.
10. Email the dev list, opening the 72 hour vote window on the RC.
11. If the vote fails or there are issues that need to be addressed, cancel, rinse and repeat.
12. Once the PMC vote passes, release the Nexus staging repo.
13. Send a RESULT email in reply to the PMC vote thread with the result.
14. Copy the RC tarballs etc to the release area.
15. Push non-RC tags to the remote repos.
16. Update the version on the branch in the repos to the next SNAPSHOT and push.
17. Update the jclouds.apache.org site.
    a. Add a new release notes file in releasenotes.
    b. Update latest_version and latest_snapshot (if necessary) in _config.yml
    c. Add the release to the DOAP file
    d. Deploy the documentation.
18. Update jclouds-examples with any relevant changes.
19. Publish the API docs for the release.
20. Mark the release as, well, released in JIRA and create the next release in JIRA, if it's not already there.
21. Send out an ANNOUNCE email to the appropriate lists (see below) to announce the release!

Prerequisites for doing a release

Apache Nexus Account and Local Configuration

See http://www.apache.org/dev/publishing-maven-artifacts.html for the details. You should already have a Nexus account created for you automatically. If you aren't able to log in to https://repository.apache.org, and you're a jclouds committer, you should probably open a JIRA with INFRA.

GPG Signing Key

- Make sure you've got a GPG signing key set up. See http://www.apache.org/dev/release-signing.html for details.
- Associate your key with your ASF account by adding the fingerprint at https://id.apache.org/.
- At least for now, also make sure your key is in people.apache.org:/www/www.apache.org/dist/jclouds/KEYS.
- If/when you can, get linked into the Apache Web of Trust.
- Optionally, you can also add your GPG passphrase to your ~/.m2/settings.xml, or enable gpg-agent for the maven-gpg-plugin, by doing something like this:

```
<profile>
  <id>apache-release</id>
  <properties>
    <gpg.useagent>true</gpg.useagent>
    <gpg.passphrase>[GPG_PASSWORD]</gpg.passphrase>
  </properties>
</profile>
```

- But remember that this is not particularly secure, so you probably should just suck it up and enter your passphrase when you kick off the build, though you can leave gpg.useagent set to true to avoid having to re-enter it for each repo.

Mailing Lists

You will need to be subscribed to all mailing lists to which announcements need to be sent.

- dev@jclouds.a.o
- user@jclouds.a.o
- announce@a.o
- jclouds-announce@meetup.com

Cutting an RC

Running the RC build

1. The repo order to run the RC build is:
   a. jclouds.git
   b. jclouds-labs.git
   c. jclouds-labs-openstack.git
   d. jclouds-labs-aws.git (if 1.7 or later)
   e. jclouds-labs-google.git (if 1.7 or later)
   f. jclouds-karaf.git
   g. jclouds-cli.git
2. For each of those repos, in that order, do the following:
   a. Check out the branch you're going to be releasing from - e.g., 1.6.x
   b. Check that your Git username and email are set to match your Apache account
   c. Check the <repo>/pom.xml and/or <repo>/project/pom.xml for any properties (typically <jclouds.version>) referencing the SNAPSHOT version we're leaving, and replace those with the release version. The following command can help finding all those versions:
      i. find . -name pom.xml -exec grep -nH '1.6.3-SNAPSHOT' {} \; | grep -v '<version>'
   d. Commit the changes, as otherwise the following step will complain about uncommitted local modifications, but don't push them.
      i. For jclouds-labs-openstack.git, first run mvn versions:update-parent -DgenerateBackupPoms=false -DparentVersion=" [1.6.3]" (or whatever the release version is) and commit the changes (don't push them yet).
   e. Run mvn --version and verify that you are using a Java 6 JDK for 1.6.x or newer releases
   f. Give your build more memory. e.g.
      i. export MAVEN_OPTS="-Xms512m -Xmx1024m -XX:PermSize=256m -XX:MaxPermSize=1024m"
   g. Prepare the release. Remember to replace "jclouds-1.6.3-rcX" and the versions with the correct tag/version for the repo/release/RC you're building. e.g.:
      i. mvn release:clean release:prepare -DreleaseVersion=1.6.3 -Dtag=jclouds-1.6.3-rcX -DdevelopmentVersion=1.6.4-SNAPSHOT - DpushChanges=false
   h. Maven will ask you some questions:
      i. There are still some remaining snapshot dependencies. Do you want to resolve them now? yes
      ii. Dependency type to resolve,: specify the selection number: 0
      iii. Which release version should it be set to? 1.6.3
      iv. What version should the dependency be reset to for development? 1.6.4-SNAPSHOT
   i. Some repositories (most notably labs and karaf) may have "missing dependency" build problems with the new version when you run release:prepare. If that happens, run mvn clean install -DskipTests with the POMs set to the release version, and then try running the prepare command again.
   j. Note that this will explicitly and specifically **not** push the tag or branch to the remote (whatever).git. That is by design, to make it easier to recover from errors and to speed up the release build. You will need to push the tag and branch upon completion of the build.
   k. Perform the release build. e.g.,: mvn clean release:perform -DconnectionUrl=scm:git:file://`pwd`/.git -Dtag=jclouds-1.6.3-rcX - Darguments="-DskipTests"
   l. We deliberately do not run the tests in the release build, since we already just ran them for the prepare build, and that just seems redundant.
   m. JARs will be deployed to repository.apache.org, in a new staging repo (if this is the first of the repos you're building) or the existing one used for the previous repos (if you've already built jclouds.git). Remember that you need to have an account on repository.apache.org and your GPG key set up (as described above) or the build will fail.
   n. For those repos where we added a commit to update <jclouds.version> and similar properties, bump the properties to the next development SNAPSHOT version and commit the change

      i. For jclouds-labs-openstack.git, this can be done by running mvn versions:update-parent -DallowSnapshots=true - DgenerateBackupPoms=false -DparentVersion="[1.6.4-SNAPSHOT]". You will need to install jclouds-project:1.6.4-SNAPSHOT in your local repository first, though.
   o. Push the tag and the release commit to the remote repos
      i. We don't push the second commit the release plugin applies, changing the version to 1.6.4-SNAPSHOT, or the commits that change <jclouds.version> and similar properties, yet. If we need to spin a second release candidate, we'll do by creating a new 1.6.4 (or whatever) branch from the last commit before the version changed to the release version, and re-run subsequent RCs there.
   p. Go to https://git-wip-us.apache.org/repos/asf?p=jclouds.git (or equivalent) and find the link to the tag you just pushed - copy that and the assoicated commit hash aside, since you'll need them for the vote email.
   q. Once you've finished building all of the repos, go to https://repository.apache.org, log in, and go to Staging Repositories. Find the open org.apache.jclouds repository, which your builds have been uploading to. Select it and close it. That'll take a little while. Then copy the URL for the repo - you'll need that shortly.

Starting the Vote

1. Checkout https://dist.apache.org/repos/dist/dev/jclouds. Create a new directory for the RC, i.e., 1.6.3-rc1.

2. Go to the candidate directory and run the following, substituting the correct version and the now-closed staging repository accordingly. This will copy down all the source tarballs, their signatures and their checksums.
   a. ./fetch_jclouds_rc.sh 1.6.3 https://repository.apache.org/content/repositories/orgapachejclouds-016 1.6.3-rc1/
3. It's not a bad idea to try taking each of the tarballs, blowing them up and then building, with a standard mvn clean install. Remember that RAT checks are done as part of the Maven build, so you don't need to run those separately.
4. svn add the new directory, as well as svn rm'ing any old RCs in the same line (i.e., any older 1.6.x RC). svn ci, and use that link under https://dist.apache.org/repos/dist/dev/jclouds/ for the release vote email.
5. Get the JIRA release notes for the release by going to https://issues.apache.org/jira/secure/ConfigureReleaseNote.jspa?projectId=12314430 and choosing HTML and the version you're releasing. Save that link for inclusion in the vote/discuss emails.
6. Send two emails to dev@jclouds.apache.org, a [VOTE] thread and a [DISCUSS] thread. Set an end time for the vote, generally 72 hours from when you're sending it (but it's generally considerate to round up, and not a bad idea to give an extra day if the release vote window goes over the weekend). Example templates for the emails are below. Replace links, versions, RC numbers, release note link, etc.

   a. [VOTE] email:

```
Subject: [VOTE] Release Apache jclouds 1.6.3 RC3
Body:
Hello,

This is the third release candidate for Apache jclouds 1.6.3.

Please use the separate [DISCUSS] thread for anything but votes.

It fixes the following issues:
https://issues.apache.org/jira/secure/ReleaseNote.jspa?version=12324412&styleName=Html&projectId=12314430

*** Please download, test and vote by Saturday, June 15th, 09:00 PDT / 12:00 EDT / 18:00 CET.

Note that we are voting upon the source (tag), binaries are provided for convenience.

Source and binary files:
https://dist.apache.org/repos/dist/dev/jclouds/jclouds-1.6.3-candidate-3

Maven staging repo:
https://repository.apache.org/content/repositories/orgapachejclouds-016

The tags to be voted upon:
 - jclouds - https://git-wip-us.apache.org/repos/asf?p=jclouds.git;a=tag;
h=8ed5571b0ecf7d79c64bc16642264684882f0311 (SHA-1: 079f9f0bcc9a4374cc73692fc16c5423b70fecbe)
 - jclouds-labs - https://git-wip-us.apache.org/repos/asf?p=jclouds-labs.git;a=tag;
h=54b084e474e0c22b539ed7cce0b0ebbc82b310cb (SHA-1: da9de656e7f9ae9706a287a45ecfe60e79833ae9)
 - jclouds-labs-aws - https://git-wip-us.apache.org/repos/asf?p=jclouds-labs-aws.git;a=tag;
h=54b084e474e0c22b539ed7cce0b0ebbc82b310cb (SHA-1: da9de656e7f9ae9706a287a45ecfe60e79833ae9)
 - jclouds-labs-google - https://git-wip-us.apache.org/repos/asf?p=jclouds-labs-google.git;a=tag;
h=54b084e474e0c22b539ed7cce0b0ebbc82b310cb (SHA-1: da9de656e7f9ae9706a287a45ecfe60e79833ae9)
 - jclouds-labs-openstack - https://git-wip-us.apache.org/repos/asf?p=jclouds-labs-openstack.git;a=tag;
h=54b084e474e0c22b539ed7cce0b0ebbc82b310cb (SHA-1: da9de656e7f9ae9706a287a45ecfe60e79833ae9)
 - jclouds-karaf - https://git-wip-us.apache.org/repos/asf?p=jclouds-karaf.git;a=tag;
h=9e1f0d14285c8edeb35499f313aa7dbfab4a86f6 (SHA-1: 4b515ee6bf7b2bcc25747b75f7ccd87e4768f66a)
 - jclouds-cli - https://git-wip-us.apache.org/repos/asf?p=jclouds-cli.git;a=tag;
h=2e3575f56de2bf67469782d94d72ce8e621ddda5 (SHA-1: adfe2e213bb0121d1faa6b7c135c092b70a93504)

jclouds KEYS file containing PGP keys we use to sign the release:
http://www.apache.org/dist/jclouds/KEYS

[ ] +1
[ ] 0  (explain why)
[ ] -1 (explain why)
```

- [DISCUSS] email

```
Subject: [DISCUSS] Release Apache jclouds 1.6.3 RC3
Body:
 This thread is for discussion of the third release candidate for Apache jclouds 1.6.3. Please use this thread
for discussion of issues uncovered in the RC, questions you may have about the RC, etc. Thank you.
```

Canceling the PMC Vote

1. If, for some reason (e.g. a -1 that is not resolved), the vote needs to be cancelled:
2. Reply to the [VOTE] thread on dev@jclouds.apache.org announcing the cancellation of the vote. Modify the template as appropriate.

```
Subject: [CANCEL][VOTE] Release Apache jclouds 1.6.3 RC3
Body:
 This vote is being cancelled due to issues uncovered during validation. A new release candidate will be created
once these issues have been addressed. Thank you.
```

1. Revert (using git revert) the release commits and push them to the remote repos.
2. If applicable, commit any additional changes required to resolved the issues identified, and push these too.
3. Drop the staging repository (take care! drop it; don't release it!)
4. Respin

Finishing the PMC Vote

1. If, after the voting window is closed, there are at least three binding +1s and no binding -1s, the PMC has signed off on the release and it can, well, be released.
2. Reply to the [VOTE] thread on dev@jclouds.apache.org announcing the closure of the vote, regardless of its outcome, and give the vote counts. Modify the template as appropriate.

```
Subject: [RESULT][VOTE] Release Apache jclouds 1.6.3 RC3
Body:
The vote is now closed, and with X binding +1s, we're ready to release/this vote has not passed.
```

1. Assuming the vote has passed, it's time to release!

Actually Releasing

1. First, in each of the repos, create and push a new tag, like jclouds-1.6.3, pointing at the same commit as the RC tag we included in the vote emails. Push those tags to the remote.
   a. git tag jclouds-1.6.3 jclouds-1.6.3-rc1
   b. git push origin jclouds-1.6.3
2. Push the second commit made by the release plugin (which updates the POM versions to the next SNAPSHOT), and commits made to update <jclouds.version> and similar properties

   a. For jclouds-labs-openstack, also run mvn versions:update-parent -DparentVersion="[1.6.4-SNAPSHOT]" or whatever the new version is, commit, push.
3. Checkout https://dist.apache.org/repos/dist/release/jclouds. Create a new directory for the release, i.e., 1.6.3. Copy the release artifacts from the RC into that directory. Point the stable symlink to that new directory. svn add the new directory and the stable symlink, as well as svn rm'ing any old releases in the same line (i.e., any older 1.6.x releases) - these are archived on archive.apache.org, but shouldn't be in the primary dist area. svn ci - your added files should show up on http://www.apache.org/dist/jclouds shortly, but will take a while to propagate to the various mirrors.
4. While waiting, go to https://repository.apache.org, log in, and go to Staging Repositories. Select the staging repo for the RC, and click Release - this will move the artifacts into the releases repo, and from there they'll be synced to Maven Central as well.
5. If you've got admin access to JIRA, mark the version you just released as, well, released, and if the next version in the train doesn't already exist, create it. If you're not in the JIRA admins for jclouds, drop an email to the dev list asking for someone to do this.
6. Once the release bits have shown up on the mirrors, send the following email, updating the content accordingly, to announce@apache.org, dev@jclouds.apache.org and user@jclouds.apache.org:

```
Subject: [ANNOUNCE] Apache jclouds 1.6.3 released
Body:
The Apache jclouds team is pleased to announce the release of jclouds 1.6.3.

Apache jclouds is an open source multi-cloud toolkit for the Java platform that gives you the freedom to create
applications that are portable across clouds while giving you full control to use cloud-specific features.

The source archives for the release are available here:
http://www.apache.org/dyn/closer.cgi/jclouds

The Maven artifacts for the release are available in Maven Central, under the org.apache.jclouds group ID.

The full change log is available here:
https://issues.apache.org/jira/browse/JCLOUDS/fixforversion/12324412

We welcome your help and feedback. For more information on how to
report problems, and to get involved, visit the project website at
http://jclouds.apache.org/

The Apache jclouds Team
```

1. And you're done!