Release Manager Process (2.4 and earlier)

THIS DOCUMENT IS PUBLIC

This document describes the step-by-step procedures for creating a release.

Cherry-picking

1. Create and populate a ``2.X.Y Blockers`` GitHub Project for each release (major or minor release). Use it to track the blocker bugs for release.
    a. Use the <2.4.x Blocker Project> as a template.
2. Leave a comment in the PR that you cherry-pick from as to what release the change will show up in example:
    a. "   Merged to devel and cherry-picked for the 2.4.0rc3 release.   "
    b. Use cherry-pick -x to automatically add the SHA to the cherrypick::

```
git cherry-pick -x SHA
```

3. When cherry-picking to minor releases, add a changelog entry for every cherrypick. Include a link to the PR or bug.
    a. For Beta periods, allow committers to use their judgement
    b. For RC periods, ask mattclay or another github admin to lock down permissions on the branch so that only the RM can cherry-pick (Github Admin Doc)

Warning: If you cherry-pick something after the last expected RC be sure to let @jladd know about the change so he can test Tower with the new code. Otherwise they expect that the last RC is the same as final. They need at least a day for testing so it might be better to create a short RC rather than release final directly.

Pre-Beta1 (Scheduling, etc)

1. Decide on the schedule for the release (from beta1 to the first rc)
2. Let Andrew Sherland know on slack what the tentative release schedule is.
3. You'll probably want to add the tentative release schedule to the email you send out announcing beta1
4. Coordinate with the Project Manager (@thaumos) about syncing open PRs from the Roadmap blocker. The roadmap blocker will have mainly features and higher level cards.
    a. Decide which open ones are to be punted vs finished.
    b. If to be finished one has a single PR, copy over.
    c. If to be finished one has multiple PRs, ask feature owner what the PRs are and move them all over to the bug
5. Get the permissions you will need
    a. Have @jimi-c create you a user on the master mirror which is:
        i. Able to ssh and sudo on ``74.207.229.77`` port 5150 (master mirror)
        ii. Has Ops in ``#ansible``
        iii. Has access to http://jenkins.ansible.eng.rdu2.redhat.com (inside the firewall)
    b. Have jimi-c give your email address permission to post to the **ansible-announce**, **ansible-project**, and **ansible-devel** mailing lists.

To make beta1 for the major release

1. Create stable-2.X and push
    a. Change version in stable-2.X:
        i. Change devel to 2.X in **docs/docsite/_themes/sphinx_rtd_theme/layout.html**
        ii. Bump the **devel** version to 2.(X+1) in the following files
            1. VERSION
            2. lib/ansible/release.py
            3. packaging/release/vars/versions.yml

b. Determine a codename for 2.(X+1) (Led Zeppelin song name) and add it to **packaging/release/vars/versions.yml**

2. Change the release information in the stable-2.4 branch in packaging/release/vars/versions.yml
3. From the stable-2.4 checkout run `` ansible-playbook packaging/release/release.yml ``
   a. Enter the release branch: stable-2.X
   b. Enter the release version: 2.X.0.0
   c. Enter the release string (ie. 0.1.beta1, or just 1 for final releases): 0.1.beta1
   d. Does this branch have git submodules? [no]: no
   e. Is this a final release (not a beta/rc)? [no]: no
   f. Push repositories upstream when done? [no]: no
4. The release has been built into: " packaging/release/ansible_release " (as long as you use the local connection on localhost) cd into that directory to do the rest of the work
5. ``git log -p`` to sanity check that the following files have the correct version info:
   a. VERSION
   b. lib/ansible/release.py
6. git push
7. Wait for shippable to run on the stable-2.X branch and check that it is good
   a. https://app.shippable.com/github/ansible/ansible/dashboard
8. While waiting, you can sync CHANGELOG.md from the stable-X.Y branch to devel (this is not a straight copy. A graphical merge tool between two checkouts works well)
9. Sync RELEASES.txt from stable-2.4 to devel
10. git push --tags
11. Push sha and tarfiles out (pre-release candidates do this manually, final releases do this in jenkins):
   a. export AUSERNAME=tkuratomi
   b. scp -P 5150 dist/* $AUSERNAME@74.207.229.77:
   c. ssh -p 5150 $AUSERNAME@74.207.229.77
   d. chmod 0644 ansible-*.tar.gz*
   e. sudo chown root:root ansible-*.tar.gz*
   f. sudo mv ~/ansible-*.tar.gz* /var/www/html/releases/ansible/
12. http://jenkins.testing.ansible.com/job/Sync_release_mirrors/
   a. Build now
13. Start the public RPM package build:
   a. http://jenkins.testing.ansible.com/job/Build_Ansible_Public_RPM_Branch/build?delay=0sec
   b. For the branch use the release tag created above.
   c. For the publish option use ``preview`` for an RC and ``release`` for a release.
14. See if the package builds worked:
   a. http://jenkins.testing.ansible.com/job/Build_Ansible_Public_RPM_Branch/
15. Send email:
   a. To ``ansible-devel``
   b. Template https://gist.github.com/abadger/3171f11b769150ae931498facd85c80d

    c. Change the versions and the sha256sum. Be sure to use https when copying links!
        i. Get the email link from the googlegroup archive https://groups.google.com/forum/#!forum/ansible-project

16. Non-release managers, please spread the rc announcement to:
    a. Working Groups
    b. In meetings
    c. On Agenda ticket
    d. Network - Network team responsibility, just remind them
    Network to code

.. note:: Step 14 and 15 should be merged with the communication section somehow

Pre RC1

- Have mattclay/github org admin Tighten permissions on the ``stable-2.X`` branch so only the release manager can merge there:
    - https://docs.google.com/document/d/1gWPtxNX4J39uIzwqQWLIsTZ1dY_AwEZzAd9bJ4XtZso/edit#
- Create the ``temp-staging-post-2.X.0`` branch for changes that are destined for the next minor release to be merged to::
    - git checkout stable-2.4
    - git checkout -b temp-staging-post-2.4.0
    - git push --set-upstream origin temp-staging-post-2.4.0

RC1

- Steps 2-12 of the beta1 release procedure
- Email to ``ansible-project@googlegroups.com`` as well as ``ansible-devel``
- Tweet the link to the release announcement in the googlegroups archive
- Send retweet request to @kaete on slack
- Change topic in ``#ansible`` irc channel similar to this:
    - Ansible - !search $topic - http://docs.ansible.com * latest releases: 2.3.2.0 / 2.2.3.0 / 2.1.6.0 * 2.4.0.0 RC1 - https://groups.google.com/forum/#!topic/ansible-project/uan6RTZ166Y

Once rc1 is released

.. note:: Check whether some of these should be done earlier: after stable-2.X has been branched, after beta1 has been released, etc.

.. note:: ``@shanemcd`` is our contact for jenkins issues

- Post a message in ``#ship_it`` in slack that the stable-2.X branch needs to be added to the tower test matrix
- Add version 2.X to docsite:
    - Make PR to https://github.com/ansible/docsite/blob/master/index.html, add a 2.X option to the dropdown
    - Hack build config at http://jenkins.testing.ansible.com/job/Build_Ansible_Docs/configure to pull from stable-2.X and rsync output to docs.ansible.com/2.X/ (TODO: who owns this, and should there be a generic stable-x.y docs build task?)
    - Re-enable automerge in devel. For instance:
        - https://github.com/ansible/ansible/pull/29086
    - Reminder to the committers: when merging PRs to devel, make sure that module PRs have the correct version added. All the ones which have already passed CI did so when it was still 2.(X-1).
        - The ansibot command ``rebuild_merge`` is helpful for this
    - Porting Guide
        - Create stub **docs/docsite/rst/porting_guides/porting_guide_2.(X+1).rst**
            - Use this template
            - Remember to update all 2.X -> old, 2.(X+1) -> new
        - Update :file:`docs/docsite/rst/porting_guides.rst` to point at the new file
        - Ensure :file:`CHANGELOG.md` contains anchor link
    - Update module's removed_in allowed list
        - Like https://github.com/ansible/ansible/pull/36191

Subsequent rcs of a major release

- All the steps of the :ref:`rc1` release.

Leading up to the final major release

- Start Google Doc draft for release email
    - This can be done in parallel to allow Core Team time to add comments
    - Include link to porting guide
    - Link CHANGELOG to the branched version
    - Add in major features from the changelog to the release announcement
    - Ping ``@dharmabumstead`` the release announcement link
- Update the changelog
    - People should have been updating the changelog as they added major features. Ping them to make sure they've done that
    - Cut and paste from the previous stable-X.Y changelog into devel so that devel has a record of what was in the X.Y minor releases.

To make a final major release

* Ping ``@nitzmahone`` or ``@jlaska`` or ``@gmainwaring`` to create an ansible2.X
  PPA for major-version-pinned releases
* In both ``devel`` and ``stable-2.4`` branch edit release status in
  :file:`docs/docsite/rst/release_and_maintenance.rst`

  * devel status is ``In development``. Update the Latest Version.
  * Current and N-1 status is ``Supported (bug **and** security fixes)`` Update current's version
    number.
  * N-2 is ``Supported (**only** security fixes)``
  * Everything older is ``Unsupported (end of life)``

* Edit the CHANGELOG file to have the release date.
* Edit packaging/release/vars/versions.yml in stable-2.4
  * Order is descending

* Steps 2-4 of the :ref:`beta1 <first_release>` section
  When inputting the information for the release.yml, be sure to specify:

  * Is this a final release (not a beta/rc)? [no]: yes

* Additional final release Sanity checks

   * :file:`RELEASES.txt`
   * :file:`packaging/rpm/ansible.spec`
   * :file:`packaging/debian/changelog`

* Steps 5-9 of the :ref:`beta1 <first_release>` section
* Build and upload the tarballs/rpm/deb packages via jenkins:

   * http://jenkins.testing.ansible.com/view/Ansible/job/Release_Ansible/
   * GIT_BRANCH=origin/tags/v2.4.3.0-1
   * CONFIRM=<check it>
   * DEB_PPA=set to ppa:ansible/ansible-2.X (upload to the version-pinned PPA first)
   * NOTE: the TAR jenkins job also generates the sha256sum files and uploads them automatically

   * http://jenkins.testing.ansible.com/job/Build_Ansible_DEB/
   * GIT_BRANCH=v2.4.3.0-1
   * OFFICIAL=yes
   * DEB_DIST=<leave as is>
   * DEB_PPA=ppa:ansible/ansible-2.4
   * If deploying a version of the current major Ansible release (eg, 2.4.x and 2.5 has not yet been released), also need to copy the packages from the version-pinned PPA to the unversioned ppa:ansible/ansible. Log into Launchpad with Jenkins credentials, choose the version-pinned PPA (eg ansible-2.4), "View package details", then "Copy packages". Select the current distro versions, set "Destination PPA" to ansible/ansible, choose "Copy existing binaries", and click "Copy packages" (this ensures the version-pinned and unversioned PPAs have the exact same content vs a rebuild of the same tag with a different signature).

* Steps 12-13 of the :ref:`beta1 <first release>` section
* Other things to check:

   * New release on https://releases.ansible.com/ansible
   * New release on https://pypi.python.org/pypi/ansible (Note, this can take up to 10 minutes to be reflected on the pypi CDN)
   * New release on the PPA
   * Old release on the PPA https://launchpad.net/~ansible/+archive/ubuntu/ansible-2.3
   * New release in the rpms directory: http://releases.ansible.com/ansible/rpm/release/

* Send email to ``ansible-announce`` and ``ansible-project`` googlegroup

   * This should have been worked on with the rest of the team prior to the final rc.
   * Get the email link from the `googlegroup archive

    <https://groups.google.com/forum/#!forum/ansible-announce>`_

* Steps 16 of the :ref:`beta1 <first_release>` section

  * NOTE: We do not tweet the final releases. Marketing handles that entirely

* Send a message to #ship_it on slack to alert tower team that the final release is out
* Email ansible-tower@redhat.com to alert tower team of final release
* Merge the ``temp-staging-post-2.4.0`` branch back into the ``stable-2.4`` branch and then remove
  the branch from the repo

  * Have a github admin (usually mattclay) relax permissions on the stable-2.4 branch so that
    anyone can commit again:

  *
https://docs.google.com/document/d/1gWPtxNX4J39uIzwqQWLIsTZ1dY_AwEZzAd9bJ4XtZso/edit#

To make a dot release
----

* In the ``stable-2.4`` branch edit :file:`packaging/release/vars/versions.yml` -- Update the release
  date. Version numbers should be listed in descending order.
* In ``stable-2.4`` branch edit release version and date in
  :file:`docs/docsite/rst/release_and_maintenance.rst`
* In the stable-X.Y branch edit the CHANGELOG file to have the release date.

* Step 4-16 of the :ref:`beta1 <first_release>` process
* Build and upload the tarballs/rpm/deb packages via jenkins:

  * http://jenkins.testing.ansible.com/view/Ansible/job/Release_Ansible/
  * GIT_BRANCH=origin/tags/v2.4.x.0-1
  * CONFIRM=<check it>
  * DEB_PPA=<leave as-is>
  * NOTE: the jenkins job does not generate sha256sum files
* Run the Jenkins DEB job to upload to the version-pinned PPA too (TODO: get this integrated into the main release job,
  but only when it's the current major release)

  * http://jenkins.testing.ansible.com/job/Build_Ansible_DEB/

* GIT_BRANCH=v2.4.x.0-1
* OFFICIAL=yes
* DEB_DIST=<leave as is>
* DEB_PPA=ppa:ansible/ansible-2.4

Building docs

.. Should this be moved to its own page? And then link to it at the points where docs should be rebuilt. At what point should the docsite be updated? Do we need to wait for that to complete before sending the final email out? We may wish to kill other Commit triggered docs builds that
are higher in the queue.

* http://jenkins.testing.ansible.com/job/Build_Ansible_Docs/build?delay=0sec (Google SSO)

  * OFFICIAL=yes
  * CLEAN: default (unchecked)
  * branch="origin/devel" or "origin/stable-2.4"
  * OLD_VERSION

    * If yes, we inhibit creating the "latest" symlink. This is intended for use when doing an official update of prior versions of the documentation.
    * http://docs.ansible.com/ansible/latest/ should point to the latest stable release.

OLD_VERSION

  * yes: For Beta/RCs and anything apart from the latest stable release
  * no: First time we want to update the latest symlink. This should only be done for vX.Y.0.0-1 build[b]

* Click Build
* If there are other GitHub documentations jobs being run, see left had "Build History", indicated by the GitHub you can click on the job then select the tiny red x from the top right which is between the progress bar and "keep this build for ever"

Communication

.. DRAFT: Who needs telling, when and how
Combine the Freeze columns? It looks like they're all the same. Maybe a
separate table of all the events that can happen

Note: Project Manager (thaumos) takes care of notifying marketing for the major releases (2.5.0, 2.6.0, etc)

Who	Final	RCs	Betas	Core Engine Freeze	Core & Curated Freeze	Community Freeze	Branch
IRC #ansible-*	x	x	x	x	x	x	x
WG Agendas	x	x		x	x	x	x
Network to code	x						
ML ansible-announce	x						
ML Ansible-project	x	x					
ML Ansible-devel	x	x	x	x	x	x	x
Twitter	x	x					
Slack #general							
Slack #shipit	x						
Slack #core_internal	x	x	x	x	x	x	x
Slack #core_networking		x	x	x	x	x	x
RH Slack #ansible							
RH Slack #ansible-network							

People: @Andrew Sherland (Program Manager for Ansible Engine)
Jladd (tower testing)
(kaete marketing)